Toric grammars : a new statistical approach to natural language modeling

Olivier Catoni **Thomas Mainguy**

Abstract: We propose a new statistical model for computational linguistics. Rather than trying to estimate directly the probability distribution of a random sentence of the language, we define a Markov chain on finite sets of sentences with many finite recurrent communicating classes and define our language model as the invariant probability measures of the chain on each recurrent communicating class. This Markov chain, that we call a communication model, recombines at each step randomly the set of sentences forming its current state, using some grammar rules. When the grammar rules are fixed and known in advance instead of being estimated on the fly, we can prove supplementary mathematical properties. In particular, we can prove in this case that all states are recurrent states, so that the chain defines a partition of its state space into finite recurrent communicating classes. We show that our approach is a decisive departure from Markov models at the sentence level and discuss its relationships with Context Free Grammars. Although the toric grammars we use are closely related to Context Free Grammars, the way we generate the language from the grammar is qualitatively different. Our communication model has two purposes. On the one hand, it is used to define indirectly the probability distribution of a random sentence of the language. On the other hand it can serve as a (crude) model of language transmission from one speaker to another speaker through the communication of a (large) set of sentences.

AMS 2000 subject classifications: Primary 62M09, 62P99, 68T50; secondary 91F20, 03B65, 91E40, 60J20.
Keywords and phrases: Probabilistic grammars, Context Free Grammars, Language model, Computational linguistics, Statistical learning, Finite state Markov chains.

1. Introduction to a new communication model

In the well known kernel approach to density estimation on a measurable space $\mathscr{X}$, the probability distribution $\mathbb{P}$ of a random variable $X \in \mathscr{X}$ is estimated from a sample $(X_1, \dots, X_n)$ of n independent copies of X as $\frac{1}{n}\sum_{i=1}^n k(X_i, \mathrm{d}x)$, where k is a suitable Markov kernel. This kernel estimate can be seen as a modification of the empirical measure $\overline{\mathbb{P}} = \frac{1}{n}\sum_{i=1}^n \delta_{X_i}$.

In the context of natural language modeling at the sentence level, $\mathscr{X}$ is the set of sentences, that is the set of sequences of words of finite length.

Finding sensible kernel estimates or sensible parametric models in this context is a challenge. Therefore, we propose here another route, that we will describe as an alternative way of producing a modification of the empirical measure. The idea is to recombine repeatedly a set of sentences. Let us describe for this a general framework, concerned with an arbitrary countable state space $\mathscr{X}$.

February 12, 2013

Let $\overline{\mathscr{P}}_n = \left\{ \frac{1}{n}\sum_{i=1}^n \delta_{x_i},\, x_i \in \mathscr{X} \right\}$ be the set of empirical measures of all possible samples of size n. Let us consider a parametric family $\{q_\theta, \theta \in \Theta\}$ of Markov kernels on $\overline{\mathscr{P}}_n$. Let us assume for simplicity that for any $P \in \overline{\mathscr{P}}_n$, the reachable set $\left\{Q \in \overline{\mathscr{P}}_n,\ \sum_{t\in\mathbb{N}} q_\theta^t(P,Q) > 0\right\}$ is finite, where q_θ^t is q_θ composed t times with itself, so that for instance $q_\theta^2(P,Q) = \sum_{P'\in\overline{\mathscr{P}}_n} q_\theta(P,P')q_\theta(P',Q)$. In this case we can define the Markov kernel

$$\widehat{q}_\theta(P,Q) = \lim_{k\to\infty} \frac{1}{k}\sum_{t=1}^k q_\theta^t(P,Q).$$

It is such that for any $P \in \overline{\mathscr{P}}_n$, $\widehat{q}_\theta(P,\cdot)$ is an invariant measure of q_θ. More generally $q_\theta\widehat{q}_\theta = \widehat{q}_\theta q_\theta = \widehat{q}_\theta$. The distribution $\widehat{q}_\theta(P,\cdot) \in \mathscr{M}_+^1\left(\overline{\mathscr{P}}_n\right)$ induces a marginal distribution $\widehat{Q}_{\theta,P}$ on $\mathscr{X}$ through the formula

$$\widehat{Q}_{\theta,P} = \sum_{Q\in\overline{\mathscr{P}}_n} \widehat{q}_\theta(P,Q)Q. \tag{1.1}$$

In this paper, we will be concerned with estimators of the form $\widehat{P} = \widehat{Q}_{\theta,\overline{\mathbb{P}}}$, if θ is fixed in advance, or of the form $\widehat{Q}_{\widehat{\theta},\overline{\mathbb{P}}}$, if $\widehat{\theta}$ is an estimator of the parameter θ depending also on $\overline{\mathbb{P}}$.

Another interpretation of our framework is to consider q_θ as a communication model. One speaker hears a set of sentences described by its empirical distribution $P \in \overline{\mathscr{P}}_n$ (which means that he will not make use of the special order in which he has heard them). He uses those sentences to learn the corresponding language. Then he teaches another speaker what he has learnt by outputting another random set of sentences, distributed according to $q_\theta(P,\cdot)$. The language model (as opposed to the communication model q_θ), is $\widehat{Q}_{\theta,P}$, the average sentence distribution along an infinite chain of communicating speakers. If we start from a recurrent state P, and we assume that θ is known, we obtain a communication model where the target sentence distribution $\widehat{Q}_{\theta,P}$ can be learnt without error from the set of sentences output by any involved speaker. Indeed $\widehat{Q}_{\theta,P} = \widehat{Q}_{\theta,Q}$ for any Q in the communicating class of P, which in this situation is also the reachable set from P.

This error free estimation behaviour is desirable for a communication model. It tells us that the language can be transmitted from speaker to speaker without distortion, a desirable feature in the case of a large number of speakers. The model may also account for weak stimulus learning, the fact that human beings learn language through a limited number of examples compared with the variety of new sentences they are able to formulate. Indeed, whereas the size of the support of $P \in \overline{\mathscr{P}}_n$, the number of sentences heard by one speaker, is constant and equal to n, the support of the language model $\widehat{Q}_{\theta,P}$ may be much larger. We will actually give a toy example where the number of sentences in the language is exponential with n.

In the language transmission interpretation, we may evaluate the interest of the model by studying whether it can model a large family of sentence distributions. This richness will depend on the number of recurrent communicating classes of the communication Markov model q_θ, since any invariant distribution $\widehat{q}_\theta(P,\cdot)$ is a convex combination of the unique invariant measures supported by each recurrent communicating class. The situation is even simpler in the case when all $P \in \overline{\mathscr{P}}_n$ are recurrent states (a fact we will be able to prove in our particular model). In this case $\widehat{q}_\theta(P,\cdot)$ is the unique invariant measure supported by the recurrent communicating class to which P belongs.

In this paper we will focus on the construction and mathematical properties of the communication model q_θ. We will also touch on the estimation problem stated in the opening of this introduction by providing some estimator $\widehat{Q}_{\widehat{\theta},\overline{\mathbb{P}}}$ where $\widehat{\theta}(\overline{\mathbb{P}})$ is an estimator of the parameter computed on the observed sample. However we will leave the mathematical properties of this estimator for further studies. We will be content with providing some promising preliminary experiments computed on a small sample and will share with the reader some qualitative explanations of its behaviour.

The parameter θ of our model will be a new kind of grammar, closely related to Context Free Grammars, but used to generate sentences in a different way.

2. Toric grammars

Now that we have explained our general framework based on a communication Markov kernel q_θ defined on empirical distributions, let us come to natural language modeling more specifically, and describe a dedicated family of kernels.

Natural language processing in linguistics has been using more and more elaborate mathematical tools (a brief presentation of some of them is given by E. Stabler in [14]). The n-gram models are widely used, although they fail to grasp the recursive nature of natural languages, and do not use the syntactic properties of sentences. Efforts have been made to improve the performance of these models, by introducing syntax, (Della Pietra et al. 1994 [9]; Roark 2001 [12]; Tan et al. 2012 [15]). One way to do this is to use Context Free Grammars, also named phrase structure grammars, introduced by N. Chomsky as possible models for the logical structure of natural languages (see for example [4–6]), and their probabilistic variants (Chi 1999 [3]). Our proposal follows this trend, but with the goal to separate ourselves from classic n-grams, seeing syntax as equivalence classes between constituents, which we try to discover.

We consider some dictionary of words D. Each statistical sample, as explained in the introduction, is made of a set of sentences. Each sentence is a sequence of words of D. The sentences may be of variable length.

To simplify notations we will use non normalized empirical measures. Thus, the state space of the communication Markov kernel q_θ will be

$$\overline{\mathscr{P}}_n = \left\{ \sum_{i=1}^{n} \delta_{s_i}, s_i \in D^+ \right\},$$

where we introduce the notation

$$D^+ = \bigcup_{j=1}^{\infty} D^j.$$

We will call $\overline{\mathscr{P}}_n$ the set of texts of length n. Let us notice that for us, texts are unordered sets of sentences. The question of generating meaningful ordered sequences of sentences is also of interest, but will not be addressed in this study.

In order to define the communication kernel, we will describe random transformations on texts, related to the notion of Context Free Grammars. Let us start with an informal presentation. The communication kernel will perform random recombinations of sentences.

Our point of view is to see a Context Free Grammar as the result of some fragmentation process applied to a set of sentences. Let us explain this on a simple example. Consider the sentence

This is my friend Peter.

Imagine we would like to represent this sentence as the result of pasting the expression *my friend* in its context, because we think language is built by cutting and pasting expressions drawn from some large set of memorized sentences. We can do this by introducing the simple Context Free Grammar

$$\boxed{0} \to \textit{This is } \boxed{1} \textit{ Peter .}$$
$$\boxed{1} \to \textit{my friend}$$

where we have used numbered framed boxes for non terminal symbols, the start symbol being $\boxed{0}$. The two rules mean that we can rewrite the start symbol $\boxed{0}$ to obtain the right-hand side of the first rule, and that we can then rewrite the non terminal symbol $\boxed{1}$ as the right-hand side of the second rule.

Since we want to see the rules of the grammar as the result of some splitting operation, we are going to use more symmetric notations. Instead of considering that we have described our original sentence with the help of two rules and two non terminal symbols $\boxed{0}$ and $\boxed{1}$, we may as well consider that we have split our original sentence into two new sentences using *three* non terminal symbols, namely $\boxed{0}\,\to$, $\boxed{1}$ and $\boxed{1}\,\to$. To emphasize this interpretation, we can adopt more symmetric notations and write these three non terminal symbols as $[_0$, $]_1$ and $[_1$. With these new notations, the representation of our original sentence is now

$$[_0 \textit{ This is }]_1 \textit{ Peter .}$$
$$[_1 \textit{ my friend}$$

In this new representation, the rewriting rules can be replaced by merge operations of the type

$$a\,]_i c + [_i\, b \;\mapsto\; abc$$

We can make this merge operations even more symmetric, if we consider that each expression can be represented by any of its circular permutations. Indeed, each expression contains exactly one non terminal symbol of the form $[_i$, and therefore is uniquely defined by any of its circular permutations (since, due to this feature, we can define the permutation in which the opening bracket $[_i$ comes first as the canonical form, and recover it from any other circular permutation). Using this convention, we can write $a\,]_i c$ as $ca\,]_i$ and describe the merge operation as

$$ca\,]_i + [_i\,b \;\mapsto\; cab,$$

or, renaming ca as a simply as

$$a]_i + [_i b \;\mapsto\; ab.$$

Let us formalize what we have explained so far. Let D be some dictionary of words (which can be for the sake of this mathematical description any finite set, representing the words of the natural language to be modeled). Let us form the symbol set $S = D \cup \big\{[_i,]_i, i \in \mathbb{N}\big\}$. Let us define the set of circular permutations of a sequence of symbols as

$$\mathfrak{S}(w_0, \dots, w_{\ell-1}) = \big\{(w_{(i+j \bmod \ell)}, i = 0, \dots, \ell-1), j = 0, \dots, \ell-1\big\},$$

so that for instance $\mathfrak{S}(w_0, w_1, w_3) = \{w_0w_1w_2, w_1w_2w_0, w_2w_0w_1\}$, and its support (the set of symbols included in the sequence) as

$$\mathrm{supp}\big(w_0, \dots, w_{\ell-1}\big) = \big\{w_0, \dots, w_{\ell-1}\big\}.$$

Let $A^+ = \bigcup_{n=1}^{+\infty} A^n$, $]_+ = \big\{\,]_i, i \in \mathbb{N} \setminus \{0\}\big\}$, and consider the set of expressions

$$\mathscr{E} = \Big\{\; e \in \mathfrak{S}\big([_i\, a\big),\; i \in \mathbb{N},\; a \in \big(D \cup]_+\big)^+ \setminus]_+ \Big\}.$$

In plain words, an expression is a circular permutation of a finite sequence of symbols starting with an opening bracket, containing no other opening bracket and not reduced to an opening bracket followed by a closing bracket.

This definition mirrors the fact that a given rule of a Context Free Grammar has exactly one $\boxed{i}\,\to$ (the left side), and the right side of the rule cannot be just a non terminal symbol $\boxed{j}$. Indeed, if we had allowed $\boxed{i} \to \boxed{j}$, or with our notations $[_i\,]_j$, we could as well have replaced i by j everywhere.

Definition 2.1
The set of toric grammars is the set $\mathfrak{G}$ of positive measures $\mathscr{G}$ on $\mathscr{E}$ with finite support such that for any circular permutation $e' \in \mathfrak{S}(e)$ of any expression $e \in \mathscr{E}$, $\mathscr{G}(e') = \mathscr{G}(e)$.

In other words, a toric grammar $\mathscr{G}$ is a positive measure with finite support on the set of expressions $\mathscr{E}$ satisfying

$$\mathscr{G}(e) = |\mathfrak{S}(e)|^{-1} \mathscr{G}\big(\mathfrak{S}(e)\big).$$

Let us remark that, in our definition of toric grammars, on top of choosing some special notations for Context Free Grammars, we also introduced positive weights, so that it is more the support of a toric grammar than the grammar itself that corresponds to the usual notion of Context Free Grammar.

The weights will serve to keep track of word frequencies through the process of splitting a set of sentences to obtain a toric grammar.

Our aim is indeed to build a toric grammar from a text. To be consistent with our definition of grammars, we will also define texts as positive measures. Let us give a formal definition. We will forget the sentence order, a text will be an unordered set of sentences with possible repetitions.

Definition 2.2
The set $\mathfrak{T}$ of texts is the set of toric grammars with integer weights supported by $\mathfrak{S}\left([_0\, D^+\right)$, that is the set of toric grammars with integer weights using only one non terminal symbol, the start symbol $[_0$.

In this definition, it should be understood that

$$[_0\, D^+ = \left\{\left([_0, w_1, \dots, w_k\right), \text{ where } k \in \mathbb{N} \setminus \{0\} \text{ and } w_i \in D, 1 \leqslant i \leqslant k\right\},$$

and that

$$\mathfrak{S}\left([_0\, D^+\right) = \bigcup_{e \in [_0\, D^+} \mathfrak{S}(e).$$

3. A roadmap towards a communication model

We will use toric grammars as intermediate steps to define the transition probabilities of our communication model on texts. To this purpose, we will first introduce some general types of transformations on toric grammars (reminding the reader that in our formalism texts are some special subset of toric grammars).

It will turn out that two types of expressions, global expressions and local expressions, will play different roles. Let us define them respectively as

$$\mathscr{E}_g = \mathscr{E} \cap \mathfrak{S}\left([_0\, S^+\right),$$
$$\mathscr{E}_\ell = \mathscr{E} \cap \mathfrak{S}\left([_+\, S^+\right),$$

where we remind that $[_+ = \left\{[_i, i \in \mathbb{N} \setminus \{0\}\right\}$ and $S^+ = \bigcup_{j=1}^\infty S^j$. Any toric grammar $\mathscr{G} \in \mathfrak{G}$ can be accordingly decomposed into $\mathscr{G} = \mathscr{G}_g + \mathscr{G}_\ell$, where $\mathscr{G}_g(A) = \mathscr{G}\left(A \cap \mathscr{E}_g\right)$ and $\mathscr{G}_\ell(A) = \mathscr{G}\left(A \cap \mathscr{E}_\ell\right)$, for any subset $A \subset \mathscr{E}$.

The transitions of the communication chain with kernel $q_\theta(\mathscr{T}, \mathscr{T}')$ will be defined in two steps. The first step consists in learning from the text $\mathscr{T}$ a toric grammar $\mathscr{G}$. To this purpose we will split the sentences of $\mathscr{T}$ into syntactic constituents. The second step consists in merging the constituents again to produce a random new text $\mathscr{T}'$. The parameter $\theta = \mathscr{R}$ of the communication kernel q_θ, will also be a toric grammar. The role of this reference grammar $\mathscr{R}$ will be to provide a stock of local expressions to be used when computing $\mathscr{G}$

from $\mathscr{T}$. We will discuss later the question of the estimation of $\mathscr{R}$ itself. For the time being, we will assume that the reference grammar $\mathscr{R}$ is a parameter of the communication chain, known to all involved speakers.

We could have defined a communication kernel $q_{\widehat{\mathscr{R}}(\mathscr{T})}(\mathscr{T}, \mathscr{T}')$, where the reference grammar $\widehat{\mathscr{R}}(\mathscr{T})$ itself is estimated at each step from the current text $\mathscr{T}$, but we would have obtained a model with weaker properties, where, in particular, all the states are not necessarily recurrent states. On the other hand, the proof that the reachable set from any starting point is finite still holds for this modified model, so that it does provide an alternative way of defining a language model as described in the introduction.

We will still need an estimator $\widehat{\mathscr{R}}(\mathscr{T})$ of the reference grammar, in order to provide a language estimator $\widehat{Q}_{\widehat{\mathscr{R}}(\mathscr{T}),\mathscr{T}'}$, where we are using the notations of eq. (1.1) on page 2. The estimation $\widehat{\mathscr{R}}(\mathscr{T})$ of the reference grammar will be achieved by running some fragmentation process on the text $\mathscr{T} \in \mathfrak{T}$.

4. Non stochastic syntax splitting and merging

Let us now describe the model, starting with the description of some non random grammar transformations. We already introduced a model for grammars that includes texts as a special case. We have now to describe how to generate a toric grammar from a text, with, or without, the help of a reference grammar to learn the local component of the grammar. The mechanism producing a grammar from a text will be some sort of random parse algorithm (or rather tentative parse algorithm).

All of this will be achieved by two transformations on toric grammars that will respectively *split* and *merge* expressions (syntagms) of a toric grammar into smaller or bigger ones. We will first describe the sets of possible splits and merges from a given grammar. This will serve as a basis to define random transitions from one grammar to another in subsequent sections.

Let us first introduce some elementary operations involving toric grammars.

$$
\begin{aligned}
e \oplus f &= \sum_{s \in \mathfrak{S}(e)} \delta_s + \sum_{s \in \mathfrak{S}(f)} \delta_s, && e, f \in \mathscr{E}, \\
e \ominus f &= \sum_{s \in \mathfrak{S}(e)} \delta_s - \sum_{s \in \mathfrak{S}(f)} \delta_s, && e, f \in \mathscr{E}, \\
\rho \otimes e &= \rho \sum_{s \in \mathfrak{S}(e)} \delta_s, && \rho \in \mathbb{R}, e \in \mathscr{E},
\end{aligned}
$$

The first operation builds a toric grammar containing expressions e and f with weights 1, and the third one builds a toric grammar containing expression e with weight ρ.

We can generalize these notations to be able to take the sum of a toric

grammar and an expression, as well as the sum of two toric grammars.

$$\mathscr{G} \oplus e = \mathscr{G} + \sum_{s \in \mathfrak{S}(e)} \delta_s, \qquad \mathscr{G} \in \mathfrak{G}, e \in \mathscr{E}$$

$$\mathscr{G} \ominus e = \mathscr{G} - \sum_{s \in \mathfrak{S}(e)} \delta_s, \qquad \mathscr{G} \in \mathfrak{G}, e \in \mathscr{E}$$

$$\mathscr{G} \oplus \mathscr{G}' = \mathscr{G} + \mathscr{G}', \qquad \mathscr{G}, \mathscr{G}' \in \mathfrak{G}.$$

With these notations, a split is described as

$$\mathscr{G}' = \mathscr{G} \ominus ab \oplus a]_i \oplus [_i b, \qquad \mathscr{G}, \mathscr{G}' \in \mathfrak{G},$$

the fact that $\mathscr{G}, \mathscr{G}' \in \mathfrak{G}$ implying that

$$i \in \mathbb{N} \setminus \{0\}, ab, a\,]_i, [_i b \in \mathscr{E} \text{ and } \mathscr{G}(ab) \geqslant 1.$$

The (partial) order relation $\mathscr{G} \leqslant \mathscr{G}'$ will also be defined by the rule

$$\mathscr{G} \leqslant \mathscr{G}' \iff \mathscr{G}' - \mathscr{G} \in \mathfrak{G},$$

or equivalently

$$\mathscr{G} \leqslant \mathscr{G}' \iff \mathscr{G}' - \mathscr{G} \in \mathscr{M}_+(\mathscr{E}).$$

Let us resume our example. Starting from the one sentence text

$$\mathscr{T} = 1 \otimes [_0 \ \textit{This is my friend Peter .}$$

we get after splitting the grammar

$$\mathscr{G} = [_0 \ \textit{This is}\ ]_1 \ \textit{Peter .} \oplus [_1 \ \textit{my friend}$$

which can also be written as

$$\mathscr{G} = \ \textit{Peter .}\ [_0 \ \textit{This is}\ ]_1 \oplus [_1 \ \textit{my friend}$$

In this example, as well as in the following, punctuation marks are treated as words, so that here the required dictionary has to include { is, my, friend, Peter, This, . }.

Splitting a sentence providing a new label for each split does not create generalization, since it allows only to merge back two expressions that came from the same split. To create a grammar capable of yielding new sentences, we need some label identification scheme. We will perform label identification through the more general process of label remapping, identification being a consequence of the fact that the map may not be one to one. Let

$$\mathfrak{F} = \big\{ f : \mathbb{N} \to \mathbb{N} \text{ such that } f(0) = 0 \big\}$$

be the set of label maps. For any symbol $]_i$ or $[_i$, $i \in \mathbb{N}$, let us define $f(\,]_i) =]_{f(i)}$ and $f([_i\,) = [_{f(i)}$. Let us also define for any word $w \in D$, $f(w) = w$ and for any expression $e = (w_0, \ldots, w_{\ell-1})$, $f(e) = (f(w_0), \ldots, f(w_{\ell-1}))$. Since any grammar $\mathscr{G} \in \mathfrak{G}$ is a measure on the set of expressions $\mathscr{E}$, we can define its image measure by f, considered as a map from $\mathscr{E}$ to $\mathscr{E}$. We will put $f(\mathscr{G}) = \mathscr{G} \circ f^{-1}$, meaning that $f(\mathscr{G})(A) = \mathscr{G}(f^{-1}(A))$, for any subset $A \subset \mathscr{E}$.

Definition 4.1
Two label maps f and $g \in \mathfrak{F}$ are said to be isomorphic if there is a one to one label map $h \in \mathfrak{F}$ such that $g = h \circ f$. In this case $h^{-1} \in \mathfrak{F}$ and $f = h^{-1} \circ g$. Two grammars $\mathscr{G}$ and $\mathscr{G}' \in \mathfrak{G}$ are said to be isomorphic if there is a one to one label map $f \in \mathfrak{F}$ such that $f(\mathscr{G}) = \mathscr{G}'$. In this case, $f^{-1}(\mathscr{G}') = \mathscr{G}$ and we will write $\mathscr{G} \equiv \mathscr{G}'$. If f and g are two isomorphic label maps, then for any toric grammar $\mathscr{G} \in \mathfrak{G}$, $f(\mathscr{G})$ and $g(\mathscr{G})$ are isomorphic grammars. In the following of this paper, to ease notations and simplify exposition, we will freely identify isomorphic label maps and isomorphic grammars and often speak of them as if they were equal.

This being put, we proceed with the introduction of a set of grammar transformations β that consist in a split with possible label remapping. The *split* will be the core component for generating a toric grammar from a text, by splitting the sentences in smaller parts (syntagms).

Definition 4.2 (Splitting rule)
For any $\mathscr{G} \in \mathfrak{G}$, let us consider

$$\beta(\mathscr{G}) = \left\{ f(\mathscr{G}'), f \in \mathfrak{F}, \mathscr{G}' \in \mathfrak{G}, \mathscr{G}' = \mathscr{G} \ominus ab \oplus a]_i \oplus [_i b \right\} \subset \mathfrak{G}.$$

Let us remark that in this definition, necessarily, $ab, a]_i, [_i b \in \mathscr{E}$, $i \in \mathbb{N} \setminus \{0\}$, $1 \otimes ab \leqslant \mathscr{G}$, and $a]_i \oplus [_i b \leqslant \mathscr{G}'$. Let us put

$$\beta^*(\mathscr{G}) = \bigcup_{n=0}^{+\infty} \underbrace{\beta \circ \cdots \circ \beta}_{n \text{ times}}(\mathscr{G}),$$

the set of grammars that can be constructed from repeated invocations of β.

Lemma 4.1
Let us recall that $S = D \cup \left\{ [_i,]_i, i \in \mathbb{N} \right\}$ and let us put $S^* = \bigcup_{n=0}^{+\infty} S^n$. For any text $\mathscr{T} \in \mathfrak{T}$, and any $\mathscr{G} \in \beta^*(\mathscr{T})$, $\mathscr{G}$ is a toric grammar with integer weights,

$$\begin{aligned} \mathscr{G}([_i S^*) &= \mathscr{G}(]_i S^*), && i \in \mathbb{N} \setminus \{0\}, \\ \mathscr{G}(wS^*) &= \mathscr{T}(wS^*), && w \in \left(D \cup \{[_0\}\right), \end{aligned}$$

and in particular

$$\begin{aligned} &\mathscr{G}([_0 S^*) = \mathscr{T}([_0 S^*), \\ &\mathscr{G}(wS^*) \leqslant \mathscr{T}(wS^*), && w \in \left(D \cup \{[_0\}\right)^+. \end{aligned}$$

This means that in any toric grammar obtained by splitting a text, the weights of expressions containing the two forms $]_i$ and $[_i$ of a label are balanced, the word frequencies are the same in the grammar and in the text, and the number of sentences contained in the text is given by the total weight of expressions containing the start symbol $[_0$ in the grammar.

Proof. For the first assertion, an induction on the number of applications of β yields the result, since

$$\mathscr{T}([_i S^*) = \mathscr{T}(]_i S^*) = 0, i \in \mathbb{N} \setminus \{0\},$$

and, for any $\mathscr{G}' = \mathscr{G} \ominus ab \oplus a\,]_i \oplus [_i\, b$, and any label $j \in \mathbb{N} \setminus \{0, i\}$,

$$\mathscr{G}'([_j S^*) = \mathscr{G}([_j S^*), \tag{4.1}$$
$$\mathscr{G}'(]_j S^*) = \mathscr{G}(]_j S^*), \tag{4.2}$$

whereas

$$\mathscr{G}'([_i S^*) = \mathscr{G}([_i S^*) + 1, \tag{4.3}$$
$$\mathscr{G}'(]_i S^*) = \mathscr{G}(]_i S^*) + 1. \tag{4.4}$$

For the second assertion, it suffices to remark that the weight of expressions beginning with a given word is invariant by application of β. Indeed, any word symbol $w \in D \cup \{[_0\}$ appears the same number of times at the beginning of an expression of $1 \otimes ab$ and of $a\,]_i \oplus [_i\, b$. □

This lemma is important, because we will subsequently impose restrictions on the splitting rule based on word frequencies. Our choice to define a new type of grammar as a positive measure on symbol sequences was made to keep track of word frequencies throughout the construction.

Let us now describe the reverse of a splitting transformation, that we will call a merge transformation. This transformation will be central in generating new texts from a toric grammar, by merging the syntagms into bigger ones, ending with a full sentence.

Definition 4.3 (Merge rule)
For any toric grammar $\mathscr{G} \in \mathfrak{G}$ we consider the following set of allowed merge transformations

$$\alpha(\mathscr{G}) = \left\{ \mathscr{G}' \in \mathfrak{G}, \mathscr{G}' = \mathscr{G} \ominus a\,]_i \ominus [_i\, b \oplus ab \right\}.$$

Let us remark that in this definition, necessarily $i \in \mathbb{N} \setminus \{0\}$, $a\,]_i, [_i\, b, ab \in \mathscr{E}$, and $a\,]_i \oplus [_i\, b \leqslant \mathscr{G}$.

The merge transformation is indeed the reverse of the *split*, in the sense that:

Lemma 4.2
For any $\mathscr{G}, \mathscr{G}' \in \beta^*(\mathfrak{T})$, $\mathscr{G}' \in \beta(\mathscr{G})$ if, and only if, there is $f \in \mathfrak{F}$ such that $f(\mathscr{G}) \in \alpha(\mathscr{G}')$.

Proof. Let us suppose that $\mathscr{G}' = f\big(\mathscr{G} \oplus a\,]_i \oplus [_i\, b \ominus ab\big)$ is in $\beta(\mathscr{G})$. Then $\mathscr{G}' = f\big(\mathscr{G}\big) \oplus f(a\,]_i) \oplus f([_i\, b) \ominus f(ab)$, so that $f(a\,]_i), f([_i\, b) \in \mathrm{supp}\big(\mathscr{G}'\big)$, $f(ab) \in \mathrm{supp}\big(f(\mathscr{G})\big)$, and consequently $f\big(a\,]_i\big), f\big([_i b\big)$ and $f(ab) \in \mathscr{E}$. Moreover $f(\mathscr{G}) = \mathscr{G}' \oplus f(a)f(b) \ominus f(a)\,]_{f(i)} \ominus [_{f(i)}\, f(b)$, so that $f(\mathscr{G}) \in \alpha(\mathscr{G}')$.

On the other hand, if for some $f \in \mathfrak{F}$, $f(\mathscr{G}) \in \alpha\big(\mathscr{G}'\big)$, $f(\mathscr{G}) = \mathscr{G}' \oplus ab \ominus a\,]_i \ominus [_i\, b$. Since $ab \in \mathrm{supp}\big(f(\mathscr{G})\big)$, there is $e \in \mathscr{E}$ such that $f(e) = ab$. But this implies

that there is $c, d \in S^+$ such that $a = f(c)$ and $b = f(d)$. We can then if needed modify f outside $\big\{j \in \mathbb{N} : [_j\, S^* \in \mathrm{supp}(\mathscr{G})\big\}$, to make sure that $i \in f(\mathbb{N})$. Let $f(j) = i$. We now get that $f\big(\mathscr{G}\big) = \mathscr{G}' \oplus f(c)f(d) \ominus f(c)\,]_{f(j)} \ominus [_{f(j)}\, f(d)$, so that $\mathscr{G}' = f\big(\mathscr{G} \oplus c\,]_j \oplus [_j\, d \ominus cd\big)$, proving that $\mathscr{G}' \in \beta(\mathscr{G})$. □

Another useful property of the merge rule is given by the following lemma:

Lemma 4.3
For any $f \in \mathfrak{F}$ and any $\mathscr{G} \in \mathfrak{G}$, $f\big(\alpha(\mathscr{G})\big) \subset \alpha\big(f(\mathscr{G})\big)$.

Proof. Indeed, any $\mathscr{G}' \in f\big(\alpha(\mathscr{G})\big)$ is of the form

$$\begin{aligned}\mathscr{G}' &= f\big(\mathscr{G} \oplus ab \ominus a\,]_i \ominus [_i\, b\big) \\ &= f\big(\mathscr{G}\big) \oplus f(a)f(b) \ominus f(a)\,]_{f(i)} \ominus [_{f(i)}\, b \in \alpha\big(f(\mathscr{G})\big).\end{aligned}$$ □

Unfortunately, repeating the merge transformation will not provide a text in all circumstances. Indeed, we can end up with some expressions of the type $[_i\, a\,]_i b$. However, since an expression is allowed to contain only one opening bracket, we are sure that $[_0 \notin \mathrm{supp}([_i\, a\,]_i\, b)$.

To continue the discussion, we will switch to a random context, where split and merge transformations are performed according to some probability measure.

5. Random split and merge processes

The grammars we described so far are obtained using splitting rules. Texts can be reconstructed using merge transformations. The splitting rules as well as the merge rules allow for multiple choices at each step. We will account for this by introducing random processes where these choices are made at random.

We will describe two types of random grammar transformations. Each of these will appear as a finite length Markov chain, where the length of the chain is given by a uniformly bounded stopping time.

- The learning process (or splitting process) will start with a text and build a grammar through iterated splits;
- the production process will start with a grammar and produce a text through iterated merge operations.

These two types of processes may be combined into a split and merge process, going back and forth between texts and toric grammars.

Let us give more formal definitions. Learning and parsing processes will be some special kinds of splitting processes, to be defined hereafter.

Definition 5.1 (Splitting process)
Given some restricted splitting rule $\beta_r : \mathfrak{G} \to 2^{\mathfrak{G}}$ from the set of grammars to the set of subsets of $\mathfrak{G}$, such that for any $\mathscr{G} \in \mathfrak{G}$, $\beta_r(\mathscr{G}) \subset \beta(\mathscr{G})$, a splitting

process *is a time homogeneous stopped Markov chain* $S_t, 0 \leqslant t \leqslant \tau$ *defined on* $\mathfrak{G}$ *such that*

$$\tau = \inf\big\{t \in \mathbb{N} : \beta_r(S_t) = \varnothing\big\},$$

$$\mathbb{P}\big(S_t = \mathscr{G}' \,|\, S_{t-1} = \mathscr{G}\big) > 0 \iff \mathscr{G}' \in \beta_r(\mathscr{G}).$$

Definition 5.2 (Production process)
A production process *is a time homogenous stopped Markov chain* $P_t, 0 \leqslant t \leqslant \sigma$ *defined on* $\mathfrak{G}$ *such that*

$$\sigma = \inf\big\{t \in \mathbb{N}, \alpha(P_t) = \varnothing\big\},$$

and

$$\mathbb{P}\big(P_t = \mathscr{G}' \,|\, P_{t-1} = \mathscr{G}\big) > 0 \iff \mathscr{G}' \in \alpha(\mathscr{G}).$$

Definition 5.3 (Split and Merge process)
Given a splitting process $S_t, t \in \mathbb{N}$ *and a production process* $P_t, t \in \mathbb{N}$*, a* split and merge process *is a Markov chain* $G_t \in \mathfrak{G}$*,* $t \in \mathbb{N}$*, with transitions*

$$\begin{aligned}
&\mathbb{P}\big(G_{2t+1} = \mathscr{G}' \,|\, G_{2t} = \mathscr{G}\big) = \mathbb{P}\big(S_\tau = \mathscr{G}' \,|\, S_0 = \mathscr{G}\big), && t \in \mathbb{N},\\
&\mathbb{P}\big(G_{2t} = \mathscr{G}' \,|\, G_{2t-1} = \mathscr{G}\big) = \mathbb{P}\big(P_\sigma = \mathscr{G}' \,|\, P_0 = \mathscr{G}, P_\sigma \in \mathfrak{T}\big), && t \in \mathbb{N} \setminus \{0\},
\end{aligned}$$

whose initial distribution is a probability measure on texts, so that almost surely $G_0 \in \mathfrak{T}$*.*

Let us remark that we have to impose the condition that $P_\sigma \in \mathfrak{T}$, because the production process does not produce a true text with probability one. On the other hand it can yield back G_{2t-2} with positive probability when started at G_{2t-1}, as will be proved later on. Therefore $\mathbb{P}(P_\sigma \in \mathfrak{T} \,|\, P_0 = \mathscr{G}) > 0$ for any $\mathscr{G}$ such that $\mathbb{P}(G_{2t-1} = \mathscr{G}) > 0$. One way to simulate $\mathbb{P}_{G_{2t} \,|\, G_{2t-1}}$ is to use a rejection method, simulating repeatedly from the production process until a true text is produced. In the experiments we made, $\mathbb{P}\big(P_\sigma \in \mathfrak{T} \,|\, P_0 = \mathscr{G}\big)$ was close to one and rejection a rare event.

Proposition 5.1
Let S_t*,* P_t *and* G_t *be a splitting process, a production process and the corresponding split and merge process, starting from* $G_0 = \mathscr{T} \in \mathfrak{T}$*. For any* $\mathscr{G} \in \mathfrak{G}$*, any* $\mathscr{T}' \in \mathfrak{T}$*, such that* $\sum_{t\in\mathbb{N}} \mathbb{P}(G_{2t+1} = \mathscr{G}) > 0$ *and* $\sum_{t\in\mathbb{N}} \mathbb{P}(G_{2t} = \mathscr{T}') > 0$*,*

$$\mathbb{P}\Big(\tau \leqslant 2\big[\mathscr{T}\big(DS^*\big) - \mathscr{T}\big([_0\, S^*)\big)\big] \,\Big|\, S_0 = \mathscr{T}'\Big) = 1, \tag{5.1}$$

$$\mathbb{P}\Big(\sigma \leqslant 2\big[\mathscr{T}\big(DS^*\big) - \mathscr{T}\big([_0\, S^*)\big)\big] \,\Big|\, P_0 = \mathscr{G}\Big) = 1. \tag{5.2}$$

In other words, the length of all the splitting and production processes involved in the split and merge process have a uniform bound, given by twice the difference between the number of words and the number of sentences in the original text.

Proof. This proof is a bit lengthy and is based on some invariants in the split and merge operations. It has been put off to appendix A.1 on page 29. □

Proposition 5.2
If G_t is a split and merge process starting almost surely from the text $G_0 = \mathscr{T} \in \mathfrak{T}$, there is a finite subset of toric grammars $\mathfrak{G}_{\mathscr{T}}$ such that with probability equal to one there is for each time t a grammar G'_t isomorphic to G_t such that $G'_t \in \mathfrak{G}_{\mathscr{T}}$. Thus, after identification of isomorphic grammars, we can analyze the split and merge process as a finite state Markov chain, since the reachable set from any starting point is finite. We should however keep in mind that the finite state space $\mathfrak{G}_{\mathscr{T}}$ depends on the initial state $\mathscr{T}$, so the state space is still infinite, although any trajectory will almost surely stay in a finite subset of reachable states.

Proof. Let us assume that the labels of $\mathscr{G}$ are taken from $[\![0, W_\ell(\mathscr{G})]\!]$, meaning that $\mathscr{G}([_i\, S^*) = 0$ for $i > W_\ell(\mathscr{G})$. This can be achieved, up to grammar isomorphisms, by applying to $\mathscr{G}$ a suitable label map.
Let us define the set of canonical expressions

$$\mathscr{E}_c = \mathscr{E} \cap \left(\bigcup_{i \in \mathbb{N}} [_i\, S^* \right),$$

and the canonical decomposition of $\mathscr{G}$

$$\mathscr{G} = \sum_{e \in \mathscr{E}_c} \mathscr{G}(e) \otimes e.$$

We see that $\mathscr{G}$ can be described by the concatenation of the canonical expressions, each repeated a number of times equal to its weight, to form a sequence of symbols of length $W_s(\mathscr{G})$. From the proof of the previous proposition, we know that

$$W_s(\mathscr{G}) \leqslant M = 5W_w(\mathscr{T}) - 3W_e(\mathscr{T}) = 5\mathscr{T}(DS^*) - 3\mathscr{T}([_0 S^*).$$

We can represent $\mathscr{G}$ by a sequence of exactly M symbols by padding with trailing $[_0$ symbols the representation described above. Let us give an example

$$\mathscr{G} = 2 \otimes [_0\, w_1\,]_1 w_2 \oplus [_1\, w_3 \oplus [_1\, w_4$$

can be coded as

$$[_0\, w_1\,]_1 w_2\, [_0\, w_1\,]_1 w_2\, [_1\, w_3\, [_1\, w_4\, [_0\, [_0\, [_0$$

in the case when $M = 15$. Let us consider the set of symbols

$$S_{\mathscr{T}} = D \cup \big\{ [_0\, , [_i\, ,]_i, 0 < i \leqslant 2\big[\mathscr{T}(DS^*) - \mathscr{T}([_0\, S^*)\big] \big\}.$$

Since $\mathscr{G}$ uses only those symbols, we see from the proposed coding of $\mathscr{G}$ that it can take at most

$$|S_{\mathscr{T}}|^M$$

different values. Since

$$|S_{\mathscr{T}}| = |D| + 1 + 4\big[\mathscr{T}(DS^*) - \mathscr{T}([_0 S^*)\big]$$

we have proved that

$$|\mathfrak{G}_{\mathscr{T}}| \leqslant \Big(|D| + 1 + 4\big[\mathscr{T}(DS^*) - \mathscr{T}([_0 S^*)\big]\Big)^{5\mathscr{T}(DS^*) - 3\mathscr{T}([_0 S^*)}.$$

Let us notice that this bound, while being finite, is very large. □

6. Splitting rules and label identification

In the previous section, we introduced some class of random processes, and studied some of their general properties. In this section, we are going to describe some more specific schemes and go further in the description of split and merge processes that can learn toric grammars in a satisfactory way.

The choice of splitting rules and label identification rules has a decisive influence on the way syntactic categories and syntactic rules are learnt by the split and merge process. While it is necessary as a starting point to consider rules learnt from the text to be parsed itself, it will also be fruitful to consider the case when a previously learnt grammar $\mathscr{R} \in \mathfrak{G}$ can be used to govern the splits.

To make things easier to grasp, let us explain on some example the basics of syntactic generalization by label identification. Let us start with the simple text with two sentences.

$$G_0 = \mathscr{T} = [_0 \text{ This is my friend Peter . } \oplus [_0 \text{ This is my neighbour John .}$$

If we split "my friend" and "my neighbour" in the two sentences using the same label, we will form after two splits the grammar

$$\begin{aligned} G_1 = [_0 \text{ This is }]_1 \text{ Peter . } \oplus [_0 \text{ This is }]_1 \text{ John .} \\ \oplus [_1 \text{ my friend } \oplus [_1 \text{ my neighbour} \end{aligned}$$

If no more splits are allowed and we therefore reached the stopping time of the splitting process, so that $\tau = 2$, we can proceed to the production process, and reach after two more steps the new text G_2 that can either be $G_2 = G_0$ or

$$G_2 = [_0 \text{ This is my neighbour Peter . } \oplus [_0 \text{ This is my friend John .}$$

Now is a good time to remind the reader of the distinction made in section 3 on page 6 about local and global expressions.

Legitimate local expressions will be provided by the reference grammar $\mathscr{R}$, whereas global expressions will be deduced from the text itself. This approach will be particularly efficient in the case when the set of local expressions is smaller than the set of global expressions.

We will need two different kinds of split processes, one to learn the reference grammar from a text and the other one to perform the first part of the transitions of the communication Markov chain.

These split processes may be viewed as performing some parsing of the text they are applied to. Here, we do not use parsing as it is usually used to discover whether a sentence is correct or not, we use it instead to discover new expressions.

We will start by defining the parsing rules to be used in the communication chain. We will call them *narrow* parsing rules. We will then proceed to the definition of a *broad* parsing rule suitable for learning the reference grammar $\widehat{\mathscr{R}}(\mathscr{T})$ from a text.

Definition 6.1
Let us define the narrow parsing rule with reference grammar $\mathscr{R}$ as

$$\beta_n\big(\mathscr{G},\mathscr{R}\big) = \Big\{ \mathscr{G}' \in \mathfrak{G} : \mathscr{G}' = \mathscr{G} \oplus a\,]_i \oplus [_i\, b \ominus ab, \\ ab \in \mathscr{E}_g,\ \mathscr{R}\big([_i\, b\big) > 0 \Big\}, \quad \mathscr{G} \in \mathfrak{G}.$$

Let us remark that, due to the definition of the set of expressions $\mathscr{E}$ and of $\mathfrak{G} \subset \mathscr{M}_+(\mathscr{E})$, the fact that $\mathscr{G}$ and $\mathscr{G}' \in \mathfrak{G}$ implies that $i \in \mathbb{N} \setminus \{0\}$ in this definition, since necessarily $a\,]_i, [_i\, b \in \mathscr{E}$. It implies also that $[_0 \in \mathrm{supp}(a)$, a condition equivalent to $ab \in \mathscr{E}_g$.

The narrow parsing rule depends on $\mathscr{R}$ only through $\mathrm{supp}(\mathscr{R}) \cap \mathscr{E}_l$.

Let us define the broad parsing rule as

$$\beta_b\big(\mathscr{G},\mathscr{R}\big) = \Big\{ \mathscr{G}' \in \mathfrak{G} : \mathscr{G}' = \mathscr{G} \oplus a\,]_i \oplus [_i\, b \ominus ab, \\ \mathscr{R}\big(a\,]_i\big) + \mathscr{R}\big([_i\, b\big) > 0, \mathscr{R}\big(aS^*\big) \leqslant \mu_1 \mathscr{R}\big([_0\, S^*\big), \\ \text{and } \mathscr{R}\big(bS^*\big) \leqslant \mu_2 \mathscr{R}\big([_0\, S^*\big) \Big\}, \qquad \mathscr{G},\mathscr{R} \in \mathfrak{G},$$

where $\mu_1, \mu_2 \in \mathbb{R}_+$ are two positive real parameters.

Since the reference grammar is under construction during broad parsing, we will mainly use this rule with $\mathscr{R} = \mathscr{G}$, as will be explained later. The same learning parameters μ_1 and μ_2 are present here and in the innovation rule to be described next. They serve to split expressions into sufficiently infrequent halves, in order to constrain the model.

Let us define now maximal sequences, a notion that will be needed to define learning rules.

Definition 6.2
Given some toric grammar $\mathscr{G}$, we will say that $a \in S^+$ is $\mathscr{G}$-maximal and write $a \in \max(\mathscr{G})$ when

$$\mathscr{G}(aS^*) > \max\big\{\mathscr{G}(awS^*), \mathscr{G}(waS^*), w \in S\big\}.$$

In other words, a is a maximal subsequence among the subsequences with the same weight in $\mathscr{G}$. Note that if a is $\mathscr{G}$-maximal, usually $\mathscr{G}(a) = 0$ (meaning that a is not an expression of the grammar, but only a subexpression) and if the grammar $\mathscr{G}$ has integer weights (which will be the case if it has been produced by a split and merge process), then $\mathscr{G}(aS^*) \geqslant 2$.

Definition 6.3 (Innovation rule)
Using the notations $[_+ = \big\{[_i\,, i \in \mathbb{N} \setminus \{0\}\big\}$ and $]_+ = \big\{\,]_i, i \in \mathbb{N} \setminus \{0\}\big\}$, let us define the innovation rule with reference grammar $\mathscr{R}$ as

$$\beta_i\big(\mathscr{G},\mathscr{R}\big) = \Big\{ \mathscr{G}' \in \mathfrak{G} : \mathscr{G}' = \mathscr{G} \oplus a\,]_i \oplus [_i\, b \ominus ab,$$

$$\mathscr{R}\big([_i S^*\big) = 0,\ \{a,b\} \cap \max(\mathscr{R}) \neq \varnothing,$$
$$\mathscr{R}(aS^*) \leqslant \mu_1\, \mathscr{R}\big([_0\, S^*\big),\ \textit{and}\ \ \mathscr{R}\big(bS^*\big) \leqslant \mu_2\, \mathscr{R}\big([_0\, S^*\big)\ \Big\}.$$

Here again, the rule will be used while learning the reference grammar with $\mathscr{R} = \mathscr{G}$.

We will now introduce a label map that identifies the labels appearing in the same context.

Definition 6.4 (Label identification through context)
Given some toric grammar $\mathscr{G} \in \mathfrak{G}$, let us consider the relation $C \in \big(\mathbb{N} \setminus \{0\}\big)^2$ defined as

$$C = \Big\{(i,j) \in \big(\mathbb{N} \setminus \{0\}\big)^2 : \sum_{a \in S^*} \mathscr{G}(a\,]_i)\, \mathscr{G}(a\,]_j) + \mathscr{G}([_i\, a)\, \mathscr{G}([_j\, a) > 0\Big\}.$$

The smallest equivalence relation containing C defines a partition of $\mathbb{N}\setminus\{0\}$ into equivalence classes. Let $(A_k)_{k\in\mathbb{N}\setminus\{0\}}$ be an arbitrary indexing of this partition. Each positive integer falls in a unique class of the partition, so that the relation $i \in A_{\underline{\chi}_{\mathscr{G}}(i)}$ defines a label map $\underline{\chi}_{\mathscr{G}} : \mathbb{N} \to \mathbb{N}$ in a non ambiguous way. The choice of the indexing of the partition $(A_k)_{k\in\mathbb{N}\setminus\{0\}}$ does not matter, since two different choices lead to two isomorphic label maps. When applying $\underline{\chi}_{\mathscr{G}}$ to $\mathscr{G}$ itself, we will use the short notation $\underline{\chi}(\mathscr{G}) \overset{\text{def}}{=} \underline{\chi}_{\mathscr{G}}(\mathscr{G})$.
Let us consider the evolution of the number of labels used by $\mathscr{G}$:

$$L(\mathscr{G}) = \big|\{i \in \mathbb{N} : \mathscr{G}(\,]_i S^*) > 0\}\big|.$$

It is easy to see that $L\big(\underline{\chi}(\mathscr{G})\big) \leqslant L\big(\mathscr{G}\big)$ and that $\underline{\chi}(\mathscr{G}) \equiv \mathscr{G}$ if and only if $L\big(\underline{\chi}_{\mathscr{G}}(\mathscr{G})\big) = L\big(\mathscr{G}\big)$, where the symbol $\equiv$ means isomorphic. Accordingly there is $k \in \mathbb{N}$ such that $\underline{\chi}^{k+1}\big(\mathscr{G}\big) \equiv \underline{\chi}^{k}\big(\mathscr{G}\big)$, and we can take it to be the smallest integer such that $L\big(\underline{\chi}^{k+1}(\mathscr{G})\big) = L\big(\underline{\chi}^{k}(\mathscr{G})\big)$. Consequently, k is such that for any $n \geqslant k$, $\underline{\chi}^{n}\big(\mathscr{G}\big) \equiv \underline{\chi}^{k}\big(\mathscr{G}\big)$. We will define $\chi(\mathscr{G}) = \underline{\chi}^{k}\big(\mathscr{G}\big)$, up to grammar isomorphisms (so that $\chi(\mathscr{G})$ belongs to $\mathfrak{G}/\!\equiv$ rather than to $\mathfrak{G}$ itself).

A characterisation in terms of more elementary label maps will be established in Proposition A.6 on page 37. This characterization provides an algorithm to compute χ in practice.

We are now ready to define a learning rule.

Definition 6.5
Let us define the learning rule

$$\beta_\ell(\mathscr{G}) = \begin{cases} \beta_i\big(\mathscr{G},\mathscr{G}\big), & \textit{when } \beta_b\big(\mathscr{G},\mathscr{G}\big) = \varnothing, \\ \big\{\chi\big(\mathscr{G}'\big) : \mathscr{G}' \in \beta_b\big(\mathscr{G},\mathscr{G}\big)\big\}, & \textit{otherwise.} \end{cases}$$

We will define two kinds of splitting processes, based on two different choices of the restricted splitting rule β_r.

Definition 6.6 (Learning process)
A learning process *is a splitting process with restricted splitting rule*

$$\beta_r(\mathscr{G}) = \beta_\ell(\mathscr{G}).$$

Definition 6.7 (Parsing process)
A parsing process *with reference grammar $\mathscr{R} \in \mathfrak{G}$ is a splitting process with restricted splitting rule*

$$\beta_r(\mathscr{G}) = \beta_n(\mathscr{G}, \mathscr{R}).$$

Before we reach the aim of this paper and describe our statistical language model, we need to explore some of the properties of the production, learning and parsing processes introduced so far.

7. Parsing and generalization

Let us introduce some notations for the output of parsing, learning and production processes.

Definition 7.1
Let S_t be a parsing process, with reference grammar $\mathscr{R} \in \mathfrak{G}$. We will use the following notation for the distribution of S_τ.

$$\mathbb{G}_{\mathscr{T},\mathscr{R}} = \mathbb{P}_{S_\tau \mid S_0 = \mathscr{T}}, \qquad \mathscr{T} \in \mathfrak{T}.$$

We will also use a short notation for the distribution of the output of a production process.

$$\mathbb{T}_{\mathscr{G}} = \mathbb{P}_{P_\sigma \mid P_0 = \mathscr{G}, P_\sigma \in \mathfrak{T}}, \qquad \mathscr{G} \in \mathfrak{G}.$$

Eventually, $\mathbb{G}_{\mathscr{T}}$ will be the probability distribution of the output of a learning process S_t, according to the definition

$$\mathbb{G}_{\mathscr{T}} = \mathbb{P}_{S_\tau \mid S_0 = \mathscr{T}}, \qquad \mathscr{T} \in \mathfrak{T}.$$

At this point we obviously may consider different notions of parsing that we have to connect together. Namely, we would like to make a link between the following statements:

- $\mathbb{T}_{\mathscr{G}}(\mathscr{T}) > 0$, the grammar $\mathscr{G}$ can produce the text $\mathscr{T}$;
- $\mathbb{G}_{\mathscr{T},\mathscr{R}}(\mathscr{G}) > 0$, the text $\mathscr{T}$ can generate the grammar $\mathscr{G}$ when parsed with the help of the grammar $\mathscr{R}$;
- $\mathbb{G}_{\mathscr{T}}(\mathscr{G}) > 0$, the grammar $\mathscr{R}$ can be learnt from the text $\mathscr{T}$.

Lemma 7.1
The previous parse notions are related in the following way. For any $\mathscr{G}, \mathscr{R} \in \mathfrak{G}$, and any $\mathscr{T} \in \mathfrak{T}$,

$$\mathbb{G}_{\mathscr{T}}(\mathscr{G}) > 0 \implies \mathbb{T}_{\mathscr{G}}(\mathscr{T}) > 0,$$
$$\mathbb{G}_{\mathscr{T},\mathscr{R}}(\mathscr{G}) > 0 \implies \mathbb{T}_{\mathscr{G}}(\mathscr{T}) > 0,$$

$$\mathbb{T}_{\mathscr{G}}(\mathscr{T}) > 0 \implies \mathbb{G}_{\mathscr{T},\mathscr{G}}(\mathscr{G}) > 0.$$

Consequently, for any $\mathscr{G}, \mathscr{R} \in \mathfrak{G}$ *such that* $\big(\mathrm{supp}(\mathscr{G}) \cap \mathscr{E}_l\big) \subset \mathrm{supp}(\mathscr{R})$*, and any* $\mathscr{T} \in \mathfrak{T}$*,*

$$\mathbb{T}_{\mathscr{G}}(\mathscr{T}) > 0 \iff \mathbb{G}_{\mathscr{T},\mathscr{R}}(\mathscr{G}) > 0.$$

Proof. This is one of the core lemmas of this work. The proof is given in appendix A.2 on page 31, on account of its length. □

It has the following important implication.

Proposition 7.2
Given a parsing process S_t *based on a reference grammar* $\mathscr{R} \in \mathfrak{G}$ *and a production process* P_t*, the corresponding split and merge process* G_t *is weakly reversible, in the sense that for any* $\mathscr{T} \in \mathfrak{T}$*, any* $\mathscr{G} \in \bigcup_{t\in\mathbb{N}} \mathrm{supp}\big(\mathbb{P}_{G_{2t+1}}\big)$*,*

$$\mathbb{P}\big(G_1 = \mathscr{G} \,\big|\, G_0 = \mathscr{T}\big) > 0 \iff \mathbb{P}\big(G_2 = \mathscr{T} \,\big|\, G_1 = \mathscr{G}\big) > 0.$$

Consequently, for any $\mathscr{T}, \mathscr{T}' \in \mathfrak{T}$ *and any* $\mathscr{G}, \mathscr{G}' \in \bigcup_{t\in\mathbb{N}} \mathrm{supp}\big(\mathbb{P}_{G_{2t+1}}\big)$*,*

$$\mathbb{P}\big(G_2 = \mathscr{T}' \,\big|\, G_0 = \mathscr{T}\big) > 0 \iff \mathbb{P}\big(G_2 = \mathscr{T} \,\big|\, G_0 = \mathscr{T}'\big) > 0,$$
$$\mathbb{P}\big(G_3 = \mathscr{G}' \,\big|\, G_1 = \mathscr{G}\big) > 0 \iff \mathbb{P}\big(G_3 = \mathscr{G} \,\big|\, G_1 = \mathscr{G}'\big) > 0.$$

In other words, the two processes G_{2t} *and* G_{2t+1} *are weakly reversible time homogeneous Markov chains. As we already proved that the set of reachable states from any starting point is finite, it shows that they are recurrent Markov chains: they partition their respective state spaces into positive recurrent communicating classes.*

Proof. Let us remark first that

$$\mathbb{P}\big(G_1 = \mathscr{G} \,\big|\, G_0 = \mathscr{T}\big) > 0 \iff \mathbb{G}_{\mathscr{T},\mathscr{R}}\big(\mathscr{G}\big) > 0$$
$$\mathbb{P}\big(G_2 = \mathscr{T} \,\big|\, G_1 = \mathscr{G}\big) > 0 \iff \mathbb{T}_{\mathscr{G}}\big(\mathscr{T}\big) > 0.$$

Moreover, since $\mathscr{G} \in \mathrm{supp}\big(\mathbb{P}_{G_{2t+1}}\big)$ for some $t \in \mathbb{N}$, there is $\mathscr{T}' \in \mathfrak{T}$ such that $\mathbb{G}_{\mathscr{T}',\mathscr{R}}(\mathscr{G}) > 0$, implying that $\mathrm{supp}\big(\mathscr{G}\big) \cap \mathscr{E}_l \subset \mathrm{supp}\big(\mathscr{R}\big)$. This ends the proof according to the last statement of the previous lemma. □

8. Expectation of a random toric grammar

In section 7 on the previous page, given some text $\mathscr{T} \in \mathfrak{T}$, we defined a random distribution on toric grammars $\mathbb{G}_{\mathscr{T}}$ that we would like to use to learn a grammar from a text. The most obvious way to do this is to draw a toric grammar at random according to the distribution $\mathbb{G}_{\mathscr{T}}$, and we already saw an algorithm, described by a Markov chain and a stopping time, to do this.

The distribution $\mathbb{G}_{\mathscr{T}}$ will be spread in general on many grammars. This is a kind of instability that we would like to avoid, if possible. A natural way to get

rid of this instability would be to simulate the expectation of $\mathbb{G}_{\mathscr{T}}$. To do this, we are facing a problem: the usual definition of the expectation of $\mathbb{G}_{\mathscr{T}}$, that is

$$\int \mathscr{G}\, \mathrm{d}\mathbb{G}_{\mathscr{T}}(\mathscr{G}),$$

although well defined from a mathemacial point of view, is a meaningless toric grammar, due to the possible fluctuations of the label mapping. To get a meaningful notion of expectation, we need to define in a meaningful way the sum of two toric grammars. We will achieve this in two steps.

Let us introduce first the *disjoint sum* of two toric grammars. We will do this with the help of two disjoint label maps. Let us define the *even* and *odd* label maps f_e and f_o as

$$f_e(i) = 2i, \qquad f_o(i) = \max\{0, 2i-1\}, \qquad i \in \mathbb{N}.$$

Definition 8.1
The disjoint sum *of two toric grammars $\mathscr{G}, \mathscr{G}' \in \mathfrak{G}$ is defined as*

$$\mathscr{G} \boxplus \mathscr{G}' = f_e(\mathscr{G}) + f_o(\mathscr{G}').$$

Definition 8.2
Given a probability measure $\mathbb{G} \in \mathscr{M}_+^1(\mathfrak{G})$ with finite support, we define the mean of $\mathbb{G}$ as

$$\oint \mathscr{G}\, \mathrm{d}\mathbb{G}(\mathscr{G}) = \chi\left(\bigboxplus_{\mathscr{G} \in \mathfrak{G}} \mathbb{G}(\mathscr{G})\, \mathscr{G} \right).$$

Lemma 8.1
If G_i is an i.i.d. sequence of random grammars distributed according to $\mathbb{G}$, then almost surely

$$\lim_{n \to +\infty} \frac{1}{n} \chi\left(\bigboxplus_{i=1}^{n} G_i \right) = \oint \mathscr{G}\, \mathrm{d}\mathbb{G}(\mathscr{G}).$$

Proof. The proof of this result is quite lengthy, and postponed till appendix A.3 on page 34. □

9. Language models

We are now ready to define the language model announced in the introduction. Given a reference grammar $\mathscr{R}$, and the corresponding split and merge process $(G_t)_{t \in \mathbb{N}}$ with reference $\mathscr{R}$, we define the communication kernel $q_{\mathscr{R}}(\mathscr{T}, \mathscr{T}')$ on $\mathfrak{T}^2$ as

$$q_{\mathscr{R}}(\mathscr{T}, \mathscr{T}') = \mathbb{P}\big(G_2 = \mathscr{T}' \,|\, G_0 = \mathscr{T}\big).$$

According to Proposition 5.2 on page 13 and Proposition 7.2 on the facing page, $q_{\mathscr{R}}$ has finite reachable sets and is weakly reversible, so that all texts $\mathscr{T} \in \mathfrak{T}$ are positive recurrent states of the communication kernel $q_{\mathscr{R}}$.

Thus to each text $\mathscr{T} \in \mathfrak{T}$ corresponds a unique invariant text distribution $\widehat{q}_{\mathscr{R}}(\mathscr{T}, \cdot)$, as explained in the introduction. As all states are positive recurrent, $\widehat{q}_{\mathscr{R}}(\mathscr{T}, \cdot)$ is the unique invariant measure of $q_{\mathscr{R}}$ on the communicating class containing $\mathscr{T}$. Moreover, from the ergodic theorem,

$$\mathbb{P}\left(\widehat{q}_{\mathscr{R}}(\mathscr{T}, \cdot) = \lim_{t\to\infty} \frac{1}{t} \sum_{j=1}^{t} \delta_{G_{2j}} \;\middle|\; G_0 = \mathscr{T} \right) = 1,$$

showing that $\widehat{q}_{\mathscr{R}}(\mathscr{T}, \cdot)$ can be computed by an almost surely convergent Monte-Carlo simulation. Eventually, from the invariant probability measure on texts $\widehat{q}_{\mathscr{R}}(\mathscr{T}, \cdot)$, we deduce a probability measure on sentences $\widehat{Q}_{\mathscr{R},\mathscr{T}}$ as explained in the introduction, according to the formula

$$\widehat{Q}_{\mathscr{R},\mathscr{T}} = \mathscr{T}\big(\lfloor_0 S^*\big)^{-1} \sum_{\mathscr{T}' \in \mathfrak{T}} \widehat{q}_{\mathscr{R}}(\mathscr{T}, \mathscr{T}')\mathscr{T}'.$$

(This is the same formula as in the introduction, taking into account the fact that texts in the support of $\widehat{q}_{\mathscr{R}}(\mathscr{T}, \cdot)$ are non normalized empirical measures with the same total mass equal to $\mathscr{T}\big(\lfloor_0 S^*\big)$, the number of sentences in the text $\mathscr{T}$.)

To obtain a true language estimator, there remains to estimate $\mathscr{R}$ by some estimator $\widehat{\mathscr{R}}(\mathscr{T})$. We will do this as described in section 8 on page 18, putting

$$\widehat{\mathscr{R}}(\mathscr{T}) = \oint \mathscr{G} \, \mathrm{d}\mathbb{G}_{\mathscr{T}}(\mathscr{G}).$$

Let us remark that, according to lemma 8.1 on the preceding page, $\widehat{\mathscr{R}}(\mathscr{T})$ can be computed from repeated simulations from the distribution $\mathbb{G}_{\mathscr{T}}$.

10. Comparison with other models

10.1. Comparison with Context Free Grammars

Given a toric grammar $\mathscr{G} \in \beta^*(\mathfrak{T})$, we may consider the split and merge process G_t with reference grammar $\mathscr{G}$ starting at $G_1 = \mathscr{G}$ (so here we start at time 1 with an initial state that is a grammar, instead of starting at time 0 with an initial state that is a text). Due to the weak reversibility of Proposition 7.2 on page 18, G_2 almost surely falls in the same recurrent communicating class of $t \mapsto G_{2t}$, and the unique invariant probability measure supported by this recurrent communicating class defines a probability measure $\overline{\mathbb{T}}_{\mathscr{G}}$ on texts, and therefore a stochastic language model. This way of defining the language generated by the grammar $\mathscr{G}$ can be compared to the usual definition of the language generated by a Context Free Grammar. Indeed, the support of $\mathscr{G}$ is a Context Free Grammar, so this is meaningful to consider the language generated by this grammar and to compare it with the support of our stochastic language model.

None of these two sets of sentences is contained in the other one. In our stochastic model, the number of times a rule can be used is bounded, so if the recursive use of some rules is possible, the deterministic language will in this sense be larger. On the other hand, the stochastic model uses both production and parsing to build new sentences, whereas the deterministic model uses only production rules. In this respect, the stochastic model may, at least in some cases, define a much broader language, as we will show on the following example.

Let us take as dictionary the set

$$D = \{+, =\} \cup [\![1, N]\!],$$

where $[\![1, N]\!] = \big\{i \in \mathbb{N}, 1 \leqslant i \leqslant N\big\}$, and consider the toric grammar

$$\mathscr{G} = N^2 \otimes \ [_0\ ]_N = N \ \oplus \bigoplus_{i=1}^{N} N \otimes \ [_i\ i \ \oplus \bigoplus_{i=2}^{N} N(i-1) \otimes \ [_i\ ]_{i-1} + 1,$$

and the text

$$\mathscr{T} = N \otimes \bigoplus_{i=1}^{N} [_0\, i \underbrace{+1 + \cdots + 1}_{N-i \text{ times}} = N.$$

It is easy to check that $\mathbb{T}_{\mathscr{G}}(\mathscr{T}) > 0$, (so that $\mathscr{G} \in \beta^*(\mathscr{T})$,) that indeed the support of $\mathscr{T}$ is the language generated by $\mathrm{supp}(\mathscr{G})$, seen as a Context Free Grammar, and that the stochastic language $\overline{\mathbb{T}}_{\mathscr{G}}$ generated by $\mathscr{G}$ is able to produce with positive probability a set of sentences

$$\mathrm{supp}^2(\overline{\mathbb{T}}_{\mathscr{G}}) \stackrel{\text{def}}{=} \bigcup_{\mathscr{T} \in \mathrm{supp}(\overline{\mathbb{T}}_{\mathscr{G}})} \mathrm{supp}(\mathscr{T}),$$

equal to

$$\mathrm{supp}^2(\overline{\mathbb{T}}_{\mathscr{G}}) = \Big\{ [_0\ x_1 + \cdots + x_i = x_{i+1} + \cdots + x_j,$$
$$1 \leqslant i < j \leqslant 2N,\ x_k \in [\![1, N]\!],\ 1 \leqslant k \leqslant j,\ \sum_{k=1}^{i} x_k = \sum_{k=i+1}^{j} x_k = N \Big\}.$$

Here, the number of sentences produced by the underlying Context Free Grammar is $|\mathrm{supp}(\mathscr{T})| = N$, whereas the number of sentences produced by our stochastic language model is $|\mathrm{supp}^2(\overline{\mathbb{T}}_{\mathscr{G}})| = 2^{2(N-1)}$. Thus, in this small example based on arithmetic expressions (admittedly closer to a computing language than it is to a natural language), our new definition of the generated language induces a huge increase in the number of generated sentences.

Note that with usual Context Free Grammar notations, $\mathrm{supp}(\mathscr{G})$ would have been described as

$$\begin{aligned} &\boxed{0} \to \boxed{N} = N \\ &\boxed{i} \to i, \qquad\qquad i = 1, \ldots, N, \end{aligned}$$

$$\boxed{i} \to \boxed{i-1} + 1, \qquad\qquad i = 2, \dots, N,$$

where $\boxed{0}$ is the start symbol and $\boxed{i}$, $i = 1, \dots, N$, are other non terminal symbols.

To count the number of elements in $\mathrm{supp}^2(\overline{\mathbb{T}}_{\mathscr{G}})$, one can remark that the number of ways N can be written as $\sum_{k=1}^{i} x_k$ with an arbitrary number of terms is also the number of increasing integer sequences $0 < s_1 < \cdots < s_{i-1} < N$ of arbitrary length, which is also the number of subsets $\{s_1, \dots, s_{i-1}\}$ of $\{1, \dots, N-1\}$, that is 2^{N-1}.

Intuitively speaking, the underlying Context Free Grammar $\mathrm{supp}(\mathscr{G})$ is limited to producing a small set of global expressions of the form $i+1+\ldots+1 = N$, whereas the stochastic language model incorporates some crude logical reasoning that is capable of deducing from them a large set of new global expressions.

Let us remark also that, when we start as here from a text made of true arithmetic statements, the language generated by our language model is also made of true arithmetic statements. This shows that our approach to language modeling is capable of some sort of logical reasoning.

10.2. Comparison with Markov models

The kind of reasoning illustrated in the previous section is related to the fact that we analyse global syntactic structures represented by the global expressions of our toric grammars.

In order to give another point of comparison, we would like in this section to make a qualitative comparison with Markov models, that do not share this feature. To make a parallel between toric grammars and Markov models, we are going to show how a Markov model could be described in terms of toric grammars and label identification rules.

To build a Markov model in our framework, we have to use a deterministic splitting (or parsing) rule. This is because in a Markov model, conditional probabilities are specified from left to right in a rigid data independent way. Let us introduce the Markov splitting rule

$$\beta_m(\mathscr{G}) = \big\{\mathscr{G}' \in \mathfrak{G},\ \mathscr{G}' = \mathscr{G} \ominus [_0\ aw\ ]_i \oplus [_0\ a\ ]_j \oplus [_j\ w\ ]_i, \\ i, j \in \mathbb{N} \setminus \{0\}, a \in D^+, w \in D, \mathscr{G}\big([_j\ S^*\big) = 0\big\}.$$

We will describe now label identification rules using concepts introduced in appendix A.3 on page 34. Let us say that the pair of labels $p \in \big(\mathbb{N} \setminus \{0\}\big)^2$ is $\mathscr{G}$-Markov if there is $w \in D$ such that $\mathscr{G}\big(w]_{p_1} S^*\big)\mathscr{G}\big(w]_{p_2} S^*\big) > 0$. Let us say that the sequence of pairs of labels $p_1, \dots, p_k$ is $\mathscr{G}$-Markov if p_j is $\xi_{p_1,\dots,p_{j-1}}\big(\mathscr{G}\big)$-Markov. It can be proved as in the case of congruent sequences that if σ is a permutation and p is $\mathscr{G}$-Markov, then $p \circ \sigma$ is also $\mathscr{G}$-Markov. It can also be proved that if p and q are maximal $\mathscr{G}$-Markov sequences, then $\xi_p \equiv \xi_q$, and therefore $\xi_p(\mathscr{G}) \equiv \xi_q(\mathscr{G})$. We will call $\xi_p(\mathscr{G}) \in \mathfrak{G}/\!\equiv$ the Markov closure of $\mathscr{G}$ and

use the notation $\xi_p(\mathscr{G}) \overset{\text{def}}{\equiv} \mu(\mathscr{G})$,where $\mu(\mathscr{G})$ is the Markov pendent of $\chi(\mathscr{G})$ in the construction of toric grammars.

Let S_t, $0 \leqslant t \leqslant \tau$ be a splitting process based on the restricted splitting rule

$$\beta_r(\mathscr{G}) = \big\{\mu(\mathscr{G}'),\ \mathscr{G}' \in \beta_m(\mathscr{G})\big\}.$$

It is not very difficult to check that the support of S_τ is contained in a single isomorphic class of grammars, so that, up to label remapping the result of this splitting process is deterministic. More specifically, starting from a text

$$\mathscr{T} = \bigoplus_{j=1}^{n} [_0\, w_1^j \dots w_{\ell(j)}^j,$$

where $w_i^j \in D \setminus \{.\}$, $1 \leqslant i < \ell(j)$, $1 \leqslant j \leqslant n$, and $w_{\ell(j)}^j = .$, $1 \leqslant j \leqslant n$ so that all sentences end with a period, we obtain a grammar isomorphic to

$$\mathscr{G} = \bigoplus_{j=1}^{n} \Bigg([_0\ w_1^j\ ]_{w_1^j} \bigoplus_{i=2}^{\ell(j)-1} [_{w_{i-1}^j} w_i^j\ ]_{w_i^j} \oplus [_{w_{\ell(j)-1}^j} w_{\ell(j)}^j \Bigg),$$

where we have used words as labels instead of integers, since in this model, due to the label identification rule, labels are functions of words (namely $]_w$ is the non terminal symbol following the word $w \in D$).

We can now define a Markov production mechanism, to replace the production process. It is described as a Markov chain X_i, $i \in \mathbb{N}$, where $X_i \in D \cup \{\Delta\}$, where $\Delta \notin D$ is a padding symbol used to embed finite sentences into infinite sequences of symbols, all equal to Δ for indices larger than the sentence length. The distribution of the Markov chain X_i is as follows. Its initial distribution is

$$\mathbb{P}(X_0 = w) = \frac{\mathscr{G}\big([_0 w]_w\big)}{\mathscr{G}\big([_0 S^*\big)},$$

and its transition probabilities are

$$\begin{aligned}
&\mathbb{P}\big(X_i = \Delta \,|\, X_{i-1} = .\big) = 1,\\
&\mathbb{P}\big(X_i = . \,|\, X_{i-1} = w\big) = \frac{\mathscr{G}\big([_w .\big)}{\mathscr{G}\big([_w S^*\big)}, && w \in D \setminus \{.\}\\
&\mathbb{P}\big(X_i = w' \,|\, X_{i-1} = w\big) = \frac{\mathscr{G}\big([_w w']_{w'}\big)}{\mathscr{G}\big([_w S^*\big)}, && w, w' \in D \setminus \{.\}.
\end{aligned}$$

Roughly speaking, the difference with the production process P_t defined previously is that in the production process the production rules are drawn at random without replacement whereas here, the production rules are drawn with replacement.

It is easy to see that the initial distribution and transition probabilities of the Markov chain X_i are the empirical initial distribution and empirical transition probabilities of the training text $\mathscr{T}$.

In conclusion, to build a Markov model using the same framework as for toric grammars, we had to modify two steps in a dramatic way:

– we had to change the splitting process, and replace the random splitting process of toric grammars with a non random splitting process which chains forward transitions in a linear way;
– we had to change in a dramatic way the label identification rule to replace the *forward and backward global condition* of toric grammars with a *backward only local condition*.

(The modification of the production process is less crucial and boils down to drawing production rules with or without replacement.)

We hope that this discussion of Markov models will help the reader realize that our model proposal is indeed really different from the Markov model at sentence level. We could have extended easily the discussion to Markov models of higher order, or to more general context tree models. We let the reader figure out the details. All these more sophisticated models show the same differences from toric grammars: a more rigid splitting process and local backward label identification rules.

11. A small experiment

Let us end this study with a small example. Here we use a small text that is meant to mimic what could be found in a tutorial to learn English as a foreign language. We have added a more elaborate sentence at the end of the text to show its impact. More systematic experiments are yet to be carried out, although the conception of this model was guided by experimental trial and errors with models starting with variable length Markov chains, before we tried global rules leading to grammars.

This is the training text $\mathscr{T}$ (each line shows an expression, starting with its weight) :

```
1 [0 He is a clever guy .
1 [0 He is doing some shopping .
1 [0 He is laughing .
1 [0 He is not interested in sports .
1 [0 He is walking .
1 [0 He likes to walk in the streets .
1 [0 I am driving a car .
1 [0 I am riding a horse too .
1 [0 I am running .
1 [0 Paul is crossing the street .
1 [0 Paul is driving a car .
1 [0 Paul is riding a horse .
1 [0 Paul is walking .
1 [0 Peter is walking .
1 [0 While I was walking , I saw Paul crossing the street .
```

And now, the new sentences produced by the model (that is by $\widehat{Q}_{\widehat{\mathscr{R}},\mathscr{T}}$, approximated on 50 iterations of the communication chain with kernel $q_{\widehat{\mathscr{R}}}$).

```
1 [0 Paul is driving a car too .
```

```
1 [0 Paul is doing some shopping .
1 [0 Paul is laughing .
1 [0 Paul is riding a horse too .
1 [0 Paul is running too .
1 [0 Paul is running .
1 [0 Paul is not interested in sports too .
1 [0 Paul is not interested in sports .
1 [0 Paul is a clever guy too .
1 [0 Paul is a clever guy .
1 [0 Paul is walking too .
1 [0 Peter is driving a car too .
1 [0 Peter is driving a car .
1 [0 Peter is doing some shopping .
1 [0 Peter is laughing .
1 [0 Peter is riding a horse too .
1 [0 Peter is riding a horse .
1 [0 Peter is running too .
1 [0 Peter is running .
1 [0 Peter is not interested in sports .
1 [0 Peter is a clever guy .
1 [0 Peter is crossing the street .
1 [0 He is driving a car too .
1 [0 He is driving a car .
1 [0 He is riding a horse too .
1 [0 He is riding a horse .
1 [0 He is running too .
1 [0 He is running .
1 [0 He is not interested in sports too .
1 [0 He is crossing the street too .
1 [0 He is crossing the street .
1 [0 He is walking too .
1 [0 I am driving a car too .
1 [0 I am doing some shopping .
1 [0 I am laughing too .
1 [0 I am laughing .
1 [0 I am riding a horse .
1 [0 I am not interested in sports .
1 [0 I am a clever guy .
1 [0 I am crossing the street too .
1 [0 I am crossing the street .
1 [0 I am walking too .
1 [0 I am walking .
1 [0 While I was driving a car , I saw Paul doing some shopping too .
1 [0 While I was driving a car , I saw Paul doing some shopping .
1 [0 While I was driving a car , I saw Paul riding a horse .
1 [0 While I was driving a car , I saw Paul crossing the street .
1 [0 While I was driving a car , I saw Paul walking .
1 [0 While I was driving a car , I saw Peter riding a horse .
1 [0 While I was doing some shopping , I saw Paul riding a horse .
1 [0 While I was doing some shopping , I saw Paul walking .
```

```
1 [0 While I was laughing too , I saw Peter crossing the street .
1 [0 While I was laughing , I saw Peter riding a horse .
1 [0 While I was riding a horse , I saw Paul driving a car too .
1 [0 While I was riding a horse , I saw Paul driving a car .
1 [0 While I was riding a horse , I saw Paul laughing .
1 [0 While I was riding a horse , I saw Paul running .
1 [0 While I was riding a horse , I saw Paul walking .
1 [0 While I was riding a horse , I saw Peter not interested in sports .
1 [0 While I was running , I saw Paul laughing .
1 [0 While I was running , I saw Paul not interested in sports .
1 [0 While I was running , I saw Paul a clever guy .
1 [0 While I was running , I saw Paul walking .
1 [0 While I was not interested in sports , I saw Paul driving a car .
1 [0 While I was not interested in sports , I saw Paul riding a horse .
1 [0 While I was a clever guy , I saw Paul running .
1 [0 While I was a clever guy , I saw Paul crossing the street .
1 [0 While I was a clever guy , I saw Paul walking .
1 [0 While I was crossing the street , I saw Paul riding a horse .
1 [0 While I was crossing the street , I saw Paul running .
1 [0 While I was crossing the street , I saw Paul crossing the street .
1 [0 While I was crossing the street , I saw Paul walking .
1 [0 While I was crossing the street , I saw Peter walking .
1 [0 While I was walking , I saw Paul driving a car .
1 [0 While I was walking , I saw Paul laughing .
1 [0 While I was walking , I saw Paul riding a horse .
1 [0 While I was walking , I saw Paul running .
1 [0 While I was walking , I saw Paul not interested in sports .
1 [0 While I was walking , I saw Paul crossing the street too .
1 [0 While I was walking , I saw Paul walking .
1 [0 While I was walking , I saw Peter not interested in sports .
1 [0 While I was walking , I saw Peter walking .
```

The reference grammar was learnt first, and was computed from 10 samples of $\mathbb{G}_{\mathscr{T}}$. (We did not normalize the weights, since we were interested in the support of the local expressions only.)

```
10 [0 He likes to walk ]6 ]3 streets .
2 [0 ]1 ]8 clever guy .
2 [0 ]1 doing some shopping .
2 [0 ]1 laughing .
2 [0 ]1 not interested ]6 sports .
2 [0 ]1 riding ]8 horse .
2 [0 ]1 riding ]8 horse ]2 .
2 [0 ]1 running .
24 [0 ]7 am ]5 .
28 [0 Paul is ]5 .
40 [0 He is ]5 .
4 [0 ]1 crossing ]3 street .
4 [0 ]1 driving ]8 car .
5 [0 ]4 is ]5 .
6 [0 ]1 walking .
```

```
7 [0 Peter is ]5 .
8 [0 While ]7 was ]5 , ]7 saw ]4 ]5 .
10 [1 He is
2 [1 Peter is
2 [1 While ]7 was ]5 , ]7 saw ]4
6 [1 ]7 am
8 [1 Paul is
2 [2 too
30 [3 the
14 [4 Paul
1 [4 Peter
16 [5 crossing ]3 street
16 [5 driving ]8 car
16 [5 riding ]8 horse
34 [5 walking
8 [5 ]5 too
8 [5 ]8 clever guy
8 [5 doing some shopping
8 [5 laughing
8 [5 not interested ]6 sports
8 [5 running
20 [6 in
50 [7 I
50 [8 a
```

Although we did not yet make the software development effort required to test large text copora, we learnt a few interesting things from what we already tried:

- As it is, the model requires the inclusion of a sufficient number of simple and redundant sentences to start generalizing. At this stage, we do not know whether this could be avoided by changing the learning rules. We made quite a few attempts in this direction. All of them resulted in the production of grammatical nonsense. Breaking the global constraints that are enforced by the model seems to have a dramatic effect on grammatical coherence. This could be a clue that these global conservation rules reflect some fundamental feature of the syntactic structure of natural languages. Including a bunch of "simple" sentences made of frequent words may be seen as introducing a pinch of supervision in the learning process.
- The constraints on subexpressions frequencies in the learning rule 6.1 (page 15) and 6.3 were added to avoid some unwanted generalizations. For instance here we took $\mu_1 \mathscr{R}\big([_0\, S^*\big) = \mu_2 \mathscr{R}\big([_0\, S^*\big) = 5$. If we had chosen 10 instead of 5, sentences of the kind

  ```
  [0 While I was walking , I saw He crossing the street .
  ```

  would have emerged, where the pronoun "He" is substituted to a noun in the wrong place. We deliberately wrote the training text in such a way that "He" is more frequent than any noun, since we expect that to be true for any reasonable large corpus. Doing so, we were able to rule out

the wrong construction by lowering the frequency constraint to avoid the unwanted substitution.

– Despite all the limitations of this small example, it shows that the model is able to find out non trivial new constructs, like

```
[0 While I was laughing too, I saw Peter crossing the street.
```

where it has discovered that "too" could be added to the subordinate clause opening the sentence. We are quite pleased to see that such things could be learnt along very general label identification rules, while all the generalized sentences remain, if not all grammatically correct, at least all grammatically plausible. Of course this judgement is purely subjective. But since we have no mathematical or otherwise quantitative definition of what natural languages are, we have to be content with a subjective evaluation of models.

Studying how this learning model scales with large corpora is still a work to be done (it will require from us that we optimize our code so that it can run efficiently on large data sets).

12. Conclusion

We have built in this paper a new statistical framework for the syntactic analysis of natural languages.

The main idea pervading our approach is that trying to estimate the distribution of an isolated random sentence is hopeless. Instead we propose to build a Markov chain on sets of sentences (called texts in this paper), with non trivial recurrent communicating classes and to define our language model as the invariant measures of this Markov chain on each of these recurrent communicating classes. At each step, the Markov chain recombines the set of sentences constituting its current state, using cut and paste operations described by grammar rules. In this way we define the probability distribution of an isolated random sentence only in an indirect way. We replace the hard question of generating a random sentence by the hopefully simpler one of recombining a set of sentences in a way that keep the desired distribution invariant.

The strong points of our approach are

– a decisive departure from Markov models that are known to fail to catch the recursive structure of natural languages;
– a new "communication model" concept that defines a Markov chain on texts and in parallel on toric grammars. This results in a new definition of the language generated by what appears as a weighted Context Free Grammar (called a toric grammar in the paper). This new perspective on language production may help to overcome the challenge of weak stimulus learning;
– in this respect, the split and merge process with reference grammar $\mathscr{R}$ is the major mathematical achievement of the paper. It has non trivial

mathematical properties proving that it can be simulated using a bounded number of operations at each step, and that the state space is divided into recurrent communicating classes each including a finite number of states;
– preliminary experiments on small corpora are encouraging. They give the (acknowledgedly subjective) feeling that the model catches the structure of the natural languages we tried (French and English). Some inflection rules and other grammatical subtleties may be missed, but experimental outputs nevertheless give us the impression that we are heading in the right direction.

On the other hand, the model needs some refinements. In particular, our proposal to build a reference grammar from a text $\mathscr{T} \in \mathfrak{T}$ through the grammar expectation

$$\widehat{\mathscr{R}}(\mathscr{T}) = \oint \mathscr{G} \, \mathrm{d}\mathbb{G}_{\mathscr{T}}(\mathscr{G})$$

is clearly only a first foray into unknown territory. We hope to be able to elaborate more on this part of our research program in the future.

Appendix A: Proofs

A.1. Bound on the length of splitting and production processes

Proof of Proposition 5.1 on page 12. Let us define the length of an expression $e \in S^k \cap \mathscr{E}$ as $\ell(e) = k$. Let us introduce some remarkable weights associated with a grammar $\mathscr{G} \in \beta^*(\mathfrak{T})$.

$$\begin{aligned} W_s(\mathscr{G}) &= \sum_{e \in \mathscr{E}} \mathscr{G}(e), \\ W_e(\mathscr{G}) &= \sum_{e \in \mathscr{E}} \mathscr{G}(e) \ell(e)^{-1}, \\ W_l(\mathscr{G}) &= \sum_{i=1}^{+\infty} \mathscr{G}([_i \, S^*), \\ W_w(\mathscr{G}) &= \sum_{w \in D} \mathscr{G}(wS^*). \end{aligned}$$

Let us define the set of canonical expressions as

$$\mathscr{E}_c = \mathscr{E} \cap \left(\bigcup_{i \in \mathbb{N}} [_i S^* \right).$$

Using previously introduced notations, we can write the grammar as

$$\mathscr{G} = \sum_{e \in \mathscr{E}_c} \mathscr{G}(e) \otimes e.$$

We will call this the canonical decomposition of $\mathscr{G}$. The two weights $W_s(\mathscr{G})$ and $W_e(\mathscr{G})$ are better understood in terms of this canonical decomposition. They can be expressed as

$$W_s(\mathscr{G}) = \sum_{e \in \mathscr{E}_c} \mathscr{G}(e)\ell(e),$$
$$W_e(\mathscr{G}) = \sum_{e \in \mathscr{E}_c} \mathscr{G}(e).$$

This shows that $W_s(\mathscr{G})$ counts the "number of symbols" in the canonical decomposition of $\mathscr{G}$, whereas $W_e(\mathscr{G})$ counts the number of expressions (that is $\mathscr{G}(\mathscr{E}_c)$, the weight put by the grammar on canonical expressions). We can also see from the definitions that $W_l(\mathscr{G})$ counts the number of canonical expressions starting with a positive (that is non terminal) label, that we will call for short the number of labels, and that $W_w(\mathscr{G})$ counts the number of words.

Since a split increases the number of canonical expressions by one, the number of symbols in canonical expressions by two, the number of labels by one, and keeps the number of words constant, whereas a merge decreases these quantities in the same proportions, the following quantities are invariant in all the toric grammars involved: for any $\mathscr{G} \in \mathfrak{G}$ such that $\sum_{t \in \mathbb{N}} \mathbb{P}\big(G_t = \mathscr{G}\big) > 0$,

$$W_s(\mathscr{G}) - 2W_e(\mathscr{G}) = W_s(\mathscr{T}) - 2W_e(\mathscr{T}),$$
$$W_e(\mathscr{G}) - W_l(\mathscr{G}) = W_e(\mathscr{T}) - W_l(\mathscr{T}) = W_e(\mathscr{T}),$$
$$W_w(\mathscr{G}) = W_w(\mathscr{T}).$$

Moreover, for the same reasons, for any $\mathscr{T}' \in \mathfrak{T}$ and $\mathscr{G} \in \mathfrak{G}$ such that $\sum_{t \in \mathbb{N}} \mathbb{P}\big(G_{2t} = \mathscr{T}'\big) > 0$ and $\sum_{t \in \mathbb{N}} \mathbb{P}\big(G_{2t+1} = \mathscr{G}\big) > 0$,

$$\mathbb{P}\Big(\tau = W_l(S_\tau) \,\Big|\, S_0 = \mathscr{T}'\Big) = 1,$$
$$\mathbb{P}\Big(\sigma = W_l(\mathscr{G}) \,\Big|\, P_0 = \mathscr{G}, P_\sigma \in \mathfrak{T}\Big) = 1.$$

Thus, we will prove the lemma if we can bound $W_l(\mathscr{G})$ (or equivalently $W_l(S_\tau)$ when $S_0 = \mathscr{T}'$, since S_τ almost surely satisfies the conditions imposed on $\mathscr{G}$). We can then remark that

$$\sum_{e \in \mathscr{E}_c} \mathscr{G}(e)\mathbb{1}\big[\ell(e) \geqslant 3\big] \leqslant \sum_{e \in \mathscr{E}_c} \mathscr{G}(e)\big[\ell(e) - 2\big] = W_s(\mathscr{G}) - 2W_e(\mathscr{G}),$$
$$\begin{aligned}\sum_{e \in \mathscr{E}_c} \mathscr{G}(e)\mathbb{1}\big[\ell(e) = 2\big] &= \sum_{e \in \mathscr{E}} \mathscr{G}(e)\mathbb{1}\big[\ell(e) = 2\big] \sum_{w \in D} \mathbb{1}\big(e \in wS^*\big) \\ &\leqslant \sum_{e \in \mathscr{E}} \mathscr{G}(e) \sum_{w \in D} \mathbb{1}\big(e \in wS^*\big) = W_w(\mathscr{G}),\end{aligned}$$

because any canonical expression of length 2 is of the form $e = [_i w$, with $i \in \mathbb{N}$ and $w \in D$, so that for any $e \in \mathscr{E}_c$ of length 2,

$$\sum_{e' \in \mathfrak{S}(e)} \sum_{w \in D} \mathbb{1}\big(e' \in wS^*\big) = 1.$$

Thus

$$W_e(\mathscr{G}) \leqslant W_w(\mathscr{G}) + W_s(\mathscr{G}) - 2W_e(\mathscr{G}),$$

and consequently we can bound $W_l(\mathscr{G})$ by the split and merge invariant bound

$$W_l(\mathscr{G}) \leqslant W_l(\mathscr{G}) - W_e(\mathscr{G}) + W_w(\mathscr{G}) + W_s(\mathscr{G}) - 2W_e(\mathscr{G}).$$

This, added to the fact that $W_l(\mathscr{T}) = 0$ and $W_s(\mathscr{T}) = W_w(\mathscr{T}) + W_e(\mathscr{T})$, proves that

$$W_l(\mathscr{G}) \leqslant 2\big[W_w(\mathscr{T}) - W_e(\mathscr{T})\big].$$

This ends the proof, since $W_w(\mathscr{T}) = \mathscr{T}(DS^*)$ and $W_e(\mathscr{T}) = \mathscr{T}([_0 S^*)$. □

A.2. Parsing Relations

Proof of lemma 7.1 on page 17. The implication

$$\mathbb{T}_{\mathscr{G}}(\mathscr{T}) > 0 \implies \mathbb{G}_{\mathscr{T},\mathscr{G}}(\mathscr{G}) > 0$$

is less trivial than it may seem. Indeed we can reverse the path of the splitting process S_t, be it a parsing or a learning process, to obtain a path followed with positive probability by the production process, but reversing the production process does not give a parsing process. Let us illustrate this difficulty on a simple example. Consider

$$\mathscr{T} = 1 \otimes [_0 abcd \quad \text{and} \quad \mathscr{G} = [_0 a]_1 \oplus [_1 b]_2 \oplus [_2 c]_3 \oplus [_3 d.$$

The production path

$$\mathscr{G}, \quad [_0\, ab\,]_2 \oplus [_2\, c\,]_3 \oplus [_3\, d, \quad [_0\, ab\,]_2 \oplus [_2\, cd, \quad \mathscr{T}$$

has positive probability. The reverse path may have a positive probability for the learning process but not for the parsing process with reference $\mathscr{G}$, since none of the expressions $[_0\, ab\,]_2$ or $[_2\, cd$ belongs to the support of $\mathscr{G}$. To parse $\mathscr{T}$ according to $\mathscr{G}$, one can instead follow with positive probability such a path as

$$\mathscr{T}, \quad [_0\, abc\,]_3 \oplus [_3\, d, \quad [_0\, ab\,]_2 \oplus [_2\, c\,]_3 \oplus [_3\, d, \quad \mathscr{G}.$$

To prove the lemma, we will have to show that it is always possible to find such an alternative parsing path. This property is fundamental to our approach, since it proves that the toric grammars we build can be used to parse the texts they can produce.

Let us start with the easiest part of the proof. Assume that $\mathbb{G}_{\mathscr{T},\mathscr{R}}(\mathscr{G}) > 0$. This means that there is a path $\mathscr{G}_0, \ldots, \mathscr{G}_k$ such that $\mathscr{G}_0 = \mathscr{T}$, $\mathscr{G}_k = \mathscr{G}$, and $\mathscr{G}_t \in \beta_n(\mathscr{G}_{t-1}, \mathscr{R})$. Anyhow it is easy to check that

$$\mathscr{G}_t \in \beta_n(\mathscr{G}_{t-1}, \mathscr{R}) \implies \mathscr{G}_{t-1} \in \alpha(\mathscr{G}_t),$$

so that the reverse path is followed with a positive probability by the production process. This means that $\mathbb{T}_{\mathscr{G}}(\mathscr{T}) > 0$.

In the case of the learning process, if $\mathbb{G}_{\mathscr{T}}\big(\mathscr{G}\big) > 0$, there is a path $\mathscr{G}_t, 0 \leqslant t \leqslant k$, such that $\mathscr{G}_t \in \beta_\ell(\mathscr{G}_{t-1})$, $\mathscr{G}_0 = \mathscr{T}$ and $\mathscr{G}_k = \mathscr{G}$, consequently there is a label map $f_t \in \mathfrak{F}$ such that $f_t(\mathscr{G}_{t-1}) \in \alpha(\mathscr{G}_t)$. We can then remark that

$$f_k \circ \cdots \circ f_t(\mathscr{G}_{t-1}) \in \alpha\big(f_k \circ \cdots \circ f_{t+1}(\mathscr{G}_t)\big),$$

because as already proved before in lemma 4.3 on page 11, $f\big(\alpha(\mathscr{G})\big) \subset \alpha\big(f(\mathscr{G})\big)$. Let us consider the path $\widetilde{\mathscr{G}_t} = f_k \circ \cdots \circ f_{k-t+1}\big(\mathscr{G}_{k-t}\big)$. It begins at $\widetilde{\mathscr{G}_0} = \mathscr{G}_k = \mathscr{G}$ and ends at $\widetilde{\mathscr{G}_k} = f_k \circ \cdots \circ f_1\big(\mathscr{G}_0\big) = \mathscr{T}$. According to the previous remark, this path is followed by the production process with positive probability, proving that $\mathbb{T}_{\mathscr{G}}(\mathscr{T}) > 0$.

Let us now come to the proof of the third implication of the lemma. For this let us assume now that $\mathbb{T}_{\mathscr{G}}(\mathscr{T}) > 0$. Consider a path $\mathscr{G}_0, \ldots, \mathscr{G}_k$ such that $\mathscr{G}_0 = \mathscr{G}, \ldots, \mathscr{G}_k = \mathscr{T}$ and $\mathscr{G}_t \in \alpha(\mathscr{G}_{t-1})$. We are going to define some *decorated* path $\widetilde{\mathscr{G}_0}, \ldots, \widetilde{\mathscr{G}_k}$ with some added parentheses. Introduce a new set of symbols $B = \big\{(_i\ ,\ )_i, i \in \mathbb{N} \setminus \{0\}\big\}$ and assume that it is disjoint from the other symbols used so far, so that $B \cap S = \varnothing$. Consider the set of toric grammars $\widetilde{\mathfrak{S}}$ based on the enlarged dictionary $D \cup B$, and the projection $\pi : \widetilde{\mathfrak{G}} \to \mathfrak{G}$ defined with the help of the canonical decomposition of toric grammars as

$$\pi\left(\sum_{e \in \widetilde{\mathscr{E}_c}} \mathscr{G}(e) \otimes e\right) = \sum_{e \in \widetilde{\mathscr{E}_c}} \mathscr{G}(e) \otimes \pi(e),$$

where $\widetilde{\mathscr{E}_c}$ is the set of canonical expressions based on the enlarged dictionary $D \cup B$, and where $\pi(e)$ is obtained by removing from the sequence of symbols e the symbols belonging to the decoration set B (that is the parentheses).

Let us put $\widetilde{\mathscr{G}_0} = \mathscr{G}$ and define $\widetilde{\mathscr{G}_t}$ for $t = 1, \ldots, k$ by induction. We will check on the go that $\pi\big(\widetilde{\mathscr{G}_t}\big) = \mathscr{G}_t$. It is obviously true for $\widetilde{\mathscr{G}_0}$, because $\widetilde{\mathscr{G}_0} \in \mathfrak{S}$, so that $\pi\big(\widetilde{\mathscr{G}_0}\big) = \widetilde{\mathscr{G}_0} = \mathscr{G}_0$. That said, let us describe the construction of $\widetilde{\mathscr{G}_t}$, assuming that $\widetilde{\mathscr{G}_{t-1}}$ is already defined, and satisfies $\pi\big(\widetilde{\mathscr{G}_{t-1}}\big) = \mathscr{G}_{t-1}$. Consider the sequence of symbols a and $b \in S^*$ and the index $i \in \mathbb{N} \setminus \{0\}$ such that

$$\mathscr{G}_t = \mathscr{G}_{t-1} \oplus ab \ominus a\,]_i \ominus [_i\, b.$$

Since $\pi\big(\widetilde{\mathscr{G}_{t-1}}\big) = \mathscr{G}_{t-1}$, and since $a\,]_i \oplus [_i\, b \leqslant \mathscr{G}_{t-1}$, there are $\widetilde{a} \in \widetilde{S}^*$ and $\widetilde{b} \in \widetilde{S}^*$ such that $\pi(\widetilde{a}) = a$, $\pi(\widetilde{b}) = b$, and $\widetilde{a}\,]_i \oplus [_i\, \widetilde{b} \leqslant \widetilde{\mathscr{G}_{t-1}}$. (The choice of $\widetilde{a}$ and $\widetilde{b}$ may not be unique, in which case we can make any arbitrary choice). Let us define

$$\widetilde{\mathscr{G}_t} = \widetilde{\mathscr{G}_{t-1}} \oplus \widetilde{a}(_i\, \widetilde{b}\,)_i \ominus \widetilde{a}\,]_i \ominus [_i\, \widetilde{b}.$$

Since $\pi\big(\widetilde{a}(_i\, \widetilde{b}\,)_i\big) = \pi\big(\widetilde{ab}\big) = ab$,

$$\pi\big(\widetilde{\mathscr{G}_t}\big) = \pi\big(\widetilde{G}_{t-1}\big) \oplus \pi\big(\widetilde{a}(_i\, b\,)_i\big) \ominus \pi\big(\widetilde{a}]_i\big) \ominus \pi\big([_i\widetilde{b}\big) = \mathscr{G}_{t-1} \oplus ab \ominus a\,]_i \ominus [_i\, b = \mathscr{G}_t,$$

where we have used the obvious fact that π is linear.

We are now going to define another mapping between grammars that allows to recover $\mathscr{G}$ from any $\widetilde{\mathscr{G}}_t$ (obviously the decorations where added to keep track of $\mathscr{G}$). Let us define $\psi : \widetilde{\mathfrak{S}}' \to \mathfrak{S}$ on the set of decorated grammars $\widetilde{\mathfrak{S}}'$ which are supported by expressions where the parentheses $(_i\,)_i$ are matched (at the same level) by the formula

$$\psi\Big(\sum_{e\in\widetilde{\mathscr{E}}_c} \widetilde{\mathscr{G}}(e)\otimes e\Big) = \sum_{e\in\widetilde{\mathscr{E}}_c} \widetilde{\mathscr{G}}(e)\psi(e),$$

where $\psi(e)$ is defined by the rules

$$\psi(e) = \begin{cases} \psi\big([_i a\,]_j c\big) + \psi\big([_j\, b\big), & \text{if } e = [_i\, a(_j\, b)_j c, \text{ with } a,b,c \in \widetilde{S}^* \\ \psi(e) = 1\otimes e, & \text{otherwise.} \end{cases}$$

It is easy to check that this definition is not ambiguous and that

$$\psi(e) = \psi'(e) \oplus \bigoplus_{(_i a)_i \in \mathrm{supp}(e)} [_i \psi'(a),$$

where $\psi'(e)$ is the expression obtained from e by replacing all the sequences between outer parentheses pairs $(_j a)_j$ by $]_j$. This is may be easier to grasp on some example:

$$\psi\big([_0 a(_1 b(_2 c)_2 d)_1 e(_3 f(_4 g)_4)_3 h\big) = [_0 a\,]_1 e\,]_3 h \oplus [_1\, b\,]_2 d \oplus [_2\, c \oplus [_3\, f\,]_4 \oplus [_4\, g.$$

It is easy to check by induction that $\widetilde{\mathscr{G}}_t \in \widetilde{\mathfrak{S}}'$. Let us check moreover that $\psi\big(\widetilde{\mathscr{G}}_t\big) = \mathscr{G}$. Indeed $\psi\big(\widetilde{\mathscr{G}}_0\big) = \psi\big(\mathscr{G}\big) = \mathscr{G}$ and

$$\psi\big(\widetilde{G}_t\big) = \psi\big(\widetilde{G}_{t-1}\big) \oplus \psi\big(\widetilde{a}(_i \widetilde{b})_i\big) \ominus \psi\big(\widetilde{a}\,]_i\big) \ominus \psi\big([_i\, \widetilde{b}\big) = \psi\big(\widetilde{G}_{t-1}\big),$$

since ψ is linear and $\psi\big(\widetilde{a}(_i\, \widetilde{b})_i\big) = \psi\big(\widetilde{a}\,]_i\big) \oplus \psi\big([_i\, b\big)$.

We are now going to define a continuation for the path $\big(\widetilde{\mathscr{G}}_t, 0 \leqslant t \leqslant k\big)$ that will bring us back to $\mathscr{G}$.

We will maintain during our inductive construction two properties:

$$\psi\big(\widetilde{\mathscr{G}}_t\big) = \mathscr{G},$$
$$\text{and} \quad \mathrm{supp}\big(\widetilde{\mathscr{G}}_t\big) \cap \widetilde{\mathscr{E}}_l \subset \mathscr{E}_l,$$

where $\widetilde{\mathscr{E}}_l$ is the set of local decorated expressions, so that

$$\widetilde{\mathscr{E}}_l = \big\{e \in \widetilde{\mathscr{E}} : [_0 \notin \mathrm{supp}(e)\big\}.$$

We already proved that the first property is satisfied by $\widetilde{\mathscr{G}}_k$. As $\pi\big(\widetilde{\mathscr{G}}_k\big) = \mathscr{G}_k = \mathscr{T}$, $\mathrm{supp}(\widetilde{\mathscr{G}}_k) \cap \widetilde{\mathscr{E}}_l = \varnothing$, so that the second condition is also satisfied. Let us assume that, for some $t > k$, $\widetilde{\mathscr{G}}_{t-1}$ has been defined and satisfies the two conditions above, and let us proceed to the construction of $\widetilde{\mathscr{G}}_t$.

As long as $\widetilde{\mathscr{G}}_{t-1} \notin \mathfrak{S}$, (and this will be the case for $t < 2k$), find some canonical expression $e \in \widetilde{\mathscr{E}}_c \setminus \mathscr{E}_c$, such that $\widetilde{\mathscr{G}}_{t-1}(e) \geqslant 1$. From our induction hypotheses, we see that necessarily $[_0 \in \mathrm{supp}(e)$. Our continuation will be such that each such expression has matching parentheses with matching labels, and we will check this on the go while building it by induction. Among those matching pairs of parentheses, there is necessarily at least one inner pair. We can for instance choose the one starting with the last opening parenthesis $(_j$ of the sequence e. This choice makes it obvious that the subsequence of e enclosed between $(_j$ and $)_j$ contains no further parentheses.

Since ψ is linear and preserves positive measures, $\mathscr{G} \ominus \psi(e) = \psi\big(\widetilde{\mathscr{G}}_{t-1}\big) \ominus \psi(e) = \psi\big(\widetilde{\mathscr{G}}_{t-1} \ominus e\big) \geqslant 0$, On the other hand, e has the form $e = [_0 a (_j b)_j c$, where $\psi(b) = b$ (since $(_j)_j$ is an inner pair of parentheses in e). As $\psi(e) = \psi\big([_0 a\,]_j c\big) + \psi\big([_j b\big)$ and $\psi\big([_j b\big) = [_j b$, this shows that $[_j\, b \leqslant \mathscr{G}$, and therefore that $\mathscr{G}([_j b) > 0$. Let us now define

$$\widetilde{\mathscr{G}}_t = \widetilde{\mathscr{G}}_{t-1} \ominus e \oplus [_j b \oplus [_0\, a\,]_j c.$$

Applying ψ to $\widetilde{\mathscr{G}}_t$, we see as previously that $\psi\big(\widetilde{\mathscr{G}}_t\big) = \psi\big(\widetilde{\mathscr{G}}_{t-1}\big) = \mathscr{G}$. As $\widetilde{\mathscr{G}}_k$ contains k pairs of parentheses, and we consume one pair at each step $t > k$, we see that $\widetilde{\mathscr{G}}_{2k}$ contains no more parentheses, so that $\widetilde{\mathscr{G}}_{2k} \in \mathfrak{G}$ and $\widetilde{\mathscr{G}}_{2k} = \psi\big(\widetilde{\mathscr{G}}_{2k}\big) = \mathscr{G}$. Let us put now $\mathscr{G}_t = \pi\big(\widetilde{\mathscr{G}}_t\big)$, for $t = k+1, \dots, 2k$. We see that

$$\mathscr{G}_t = \mathscr{G}_{t-1} \ominus [_0\, abc \oplus [_0\, a\,]_j c \oplus [_j\, b,$$

where $[_0\, abc, [_0\, a\,]_j c \in \mathscr{E}$ and $\mathscr{G}\big([_j\, b\big) > 0$, so that $\mathscr{G}_t \in \beta_n(\mathscr{G}_{t-1}, \mathscr{G})$, therefore $\mathscr{G}_k = \mathscr{T}, \dots, \mathscr{G}_{2k} = \mathscr{G}$ is a path of positive probability under the parsing process with reference $\mathscr{G}$, leading from $\mathscr{T}$ to $\mathscr{G}$, in other words, $\mathbb{G}_{\mathscr{T},\mathscr{G}}(\mathscr{G}) > 0$ as required. □

A.3. Convergence to the expectation of a random toric grammar, proof of lemma 8.1 on page 19

The proof of this results is based on the fact that the operation

$$\big(\mathscr{G}, \mathscr{G}'\big) \mapsto \chi\big(\mathscr{G} \boxplus \mathscr{G}'\big)$$

is associative.

Let us begin the proof by several definitions and lemmas.

For any grammar $\mathscr{G} \in \mathfrak{G}$ and any pair of indices $p = (p^1, p^2) \in \big(\mathbb{N} \setminus \{0\}\big)^2$, we will say that p is $\mathscr{G}$-congruent when there is $a \in S^*$ such that $\mathscr{G}(a]_{p^1})\mathscr{G}(a\,]_{p^2}) > 0$ or $\mathscr{G}([_{p^1}\, a)\mathscr{G}([_{p^2}\, a) > 0$.

Let us define the label map ξ_p as

$$\xi_p(i) = \begin{cases} i, & \text{when } i \notin \{p^1, p^2\}, \\ \min\{p^1, p^2\}, & \text{when } i \in \{p^1, p^2\}. \end{cases}$$

For any sequence $p_1, \dots, p_k \in (\mathbb{N} \setminus \{0\})^{2k}$ of pairs of indices, let us define the label map $\xi_{p_1,\dots,p_k}$ as

$$\xi_{p_1,\dots,p_k} = \xi_{\xi_{p_1,\dots,p_{k-1}}(p_k)} \circ \xi_{p_1,\dots,p_{k-1}},$$

where $f\big((i,j)\big) = (f(i), f(j))$, for any $(i,j) \in \big(\mathbb{N} \setminus \{0\}\big)^2$.

Let us say that $(p_1, \dots, p_k) \in \big(\mathbb{N} \setminus \{0\}\big)^{2k}$ is $\mathscr{G}$-congruent, if $\xi_{p_1,\dots,p_{j-1}}(p_j)$ is $\xi_{p_1,\dots,p_{j-1}}\big(\mathscr{G}\big)$-congruent for any $j \leqslant k$, and that it is maximal $\mathscr{G}$-congruent is it is $\mathscr{G}$-congruent and any $\mathscr{G}$-congruent sequence of the form $(p_1, \dots, p_k, p_{k+1})$ is such that

$$\xi_{p_1,\dots,p_k}(p^1_{k+1}) = \xi_{p_1,\dots,p_k}(p^2_{k+1}),$$

or equivalently such that $\xi_{p_1,\dots,p_{k+1}} = \xi_{p_1,\dots,p_k}$.

Lemma A.1
For any sequence $(p_1, \dots, p_\ell) \in \Big(\mathbb{N} \setminus \{0\}\Big)^{2\ell}$, *for any* $k < \ell$,

$$\xi_{p_1,\dots,p_\ell} = \xi_{\xi_{p_1,\dots,p_k}(p_{k+1},\dots,p_\ell)} \circ \xi_{p_1,\dots,p_k}.$$

Proof. By induction on ℓ for k fixed. This is true from the definition for $\ell = k+1$. Assuming we have established the lemma for $\ell - 1$, we can write

$$\begin{aligned}
\xi_{p_1,\dots,p_\ell} &= \xi_{\xi_{p_1,\dots,p_{\ell-1}}(p_\ell)} \circ \xi_{p_1,\dots,p_{\ell-1}} \\
&= \xi_{\xi_{\xi_{p_1,\dots,p_k}(p_{k+1},\dots,p_{\ell-1}) \circ \xi_{p_1,\dots,p_k}(p_\ell)} \circ \xi_{\xi_{p_1,\dots,p_k}(p_{k+1},\dots,p_{\ell-1})} \circ \xi_{p_1,\dots,p_k} \\
&= \xi_{\xi_{p_1,\dots,p_k}(p_{k+1},\dots,p_\ell)} \circ \xi_{p_1,\dots,p_k}, \qquad \square
\end{aligned}$$

Lemma A.2
For any permutation σ *of* $\{1, \dots, k\}$,

$$\xi_{p_1,\dots,p_k} \equiv \xi_{p_{\sigma(1)},\dots,p_{\sigma(k)}}$$

Proof. Let us consider the smallest equivalence relation containing the set $\{p_1, \dots, p_k\}$. Let $\overline{\pi}_k : \mathbb{N} \setminus \{0\} \to \mathscr{C}$ be the corresponding projection of each label to its component. Let us befine the label map π_k by $\pi_k(0) = 0$ and

$$\pi_k(i) = \min \overline{\pi}_k(i), \qquad i > 0.$$

We are going to prove by induction on k that $\xi_{p_1,\dots,p_k} \equiv \pi_k$. Since π_k is invariant by permutation of the sequence $(p_1, \dots, p_k)$, this will prove the lemma.

Let us remark now that $\xi_{p_1,\dots,p_k} \equiv \pi_k$ if and only if

$$\xi_{p_1,\dots,p_k}(i) = \xi_{p_1,\dots,p_k}(j) \iff \pi_k(i) = \pi_k(j), \qquad i,j > 0.$$

So we are going to prove this equivalence. It is easy to see from the previous lemma that for any integer $m = 1, \dots, k$,

$$\xi_{p_1,\dots,p_k}(p^1_m) = \xi_{p_1,\dots,p_k}(p^2_m). \tag{A.1}$$

Indeed,

$$\xi_{p_1,\dots,p_k} = \xi_{\xi_{p_1,\dots,p_m}(p_{m+1},\dots,p_k)} \circ \xi_{\xi_{p_1,\dots,p_{m-1}}(p_m)} \circ \xi_{p_1,\dots,p_{m-1}},$$

so that, changing p_m for $\xi_{p_1,\dots,p_{m-1}}(p_m)$, we are back to proving the result when $m = k = 1$, where it is obvious from the definitions.

Now, eq. (A.1) on the previous page and the minimality of π_k implies that

$$\pi_k(i) = \pi_k(j) \implies \xi_{p_1,\dots,p_k}(i) = \xi_{p_1,\dots,p_k}(j), \qquad i,j > 0.$$

Let us assume conversely that $\xi_{p_1,\dots,p_k}(i) = \xi_{p_1,\dots,p_k}(j)$ and let

$$m = \min\bigl\{\ell : \xi_{p_1,\dots,p_\ell}(i) = \xi_{p_1,\dots,p_\ell}(j)\bigr\}.$$

Since $\xi_{p_1,\dots,p_m} = \xi_{\xi_{p_1,\dots,p_{m-1}}(p_m)} \circ \xi_{p_1,\dots,p_{m-1}}$, we see that necessarily

$$\xi_{p_1,\dots,p_{m-1}}\bigl(\{i,j\}\bigr) = \xi_{p_1,\dots,p_{m-1}}\bigl(\{p_m^1, p_m^2\}\bigr),$$

and that this set contains two distinct elements. Exchanging the role of i and j if necessary, we can assume without loss of generality that

$$\xi_{p_1,\dots,p_{m-1}}\bigl((i,j)\bigr) = \xi_{p_1,\dots,p_{m-1}}\bigl(p_m\bigr).$$

From the induction hypothesis, this implies that $\pi_{m-1}\bigl((i,j)\bigr) = \pi_{m-1}(p_m)$. Since the equivalence relation defined by π_{m-1} is a subset of the equivalence relation defined by π_k, this implies that $\pi_k\bigl((i,j)\bigr) = \pi_k(p_m)$. Since moreover $\pi_k(p_m^1) = \pi_k(p_m^2)$, this implies that $\pi_k(i) = \pi_k(j)$. □

Lemma A.3
For any $f \in \mathfrak{F}$, any sequence of pairs of positive labels $p_1, \dots, p_k$, there is a label map $g \in \mathfrak{F}$ such that

$$\xi_{f(p_1,\dots,p_k)} \circ f = g \circ \xi_{p_1,\dots,p_k}.$$

Proof. We have to prove that

$$\xi_{p_1,\dots,p_k}(i) = \xi_{p_1,\dots,p_k}(j) \implies \xi_{f(p_1,\dots,p_k)} \circ f(i) = \xi_{f(p_1,\dots,p_k)} \circ f(j), \qquad i,j > 0.$$

From the proof of the previous lemma, it is enough to check that the right-hand side holds when $(i,j) = p_m$, $m = 1, \dots, k$, which is then obvious. □

Lemma A.4
If $f \in \mathfrak{F}$ and $(p_1, \dots, p_k)$ is $\mathscr{G}$-congruent, then $(f(p_1), \dots, f(p_k))$ is also $f(\mathscr{G})$-congruent.

Proof. Assume that for some $a \in S^*$

$$\xi_{p_1,\dots,p_{m-1}}\bigl(\mathscr{G}\bigr)\bigl(a\,\rfloor_{\xi_{p_1,\dots,p_{m-1}}(p_m^1)}\bigr) > 0.$$

Then, $\xi_{f(p_1,\dots,p_{m-1})} \circ f = g \circ \xi_{p_1,\dots,p_{m-1}}$, and

$$\begin{aligned}\xi_{f(p_1,\dots,p_{m-1})} \circ f\big(\mathscr{G}\big)\big(g(a)\,]_{\xi_{f(p_1,\dots,p_{m-1})}\circ f(p^1_m)}\big) \\ = g \circ \xi_{p_1,\dots,p_{m-1}}\big(\mathscr{G}\big)\big(g(a)\,]_{g\circ\xi_{p_1,\dots,p_{m-1}}(p^1_m)}\big) \\ = \xi_{p_1,\dots,p_m}\big(\mathscr{G}\big)\Big(g^{-1}\circ g\big(a\,]_{\xi_{p_1,\dots,p_{m-1}}(p^1_m)}\big)\Big) \\ \geqslant \xi_{p_1,\dots,p_{m-1}}\big(\mathscr{G}\big)\big(a\,]_{\xi_{p_1,\dots,p_{m-1}}(p^1_m)}\big) > 0.\end{aligned}$$

The same is true when p^1_m is replaced with p^2_m and when $a\,]_{\xi_{p_1,\dots,p_{m-1}(p^1_m)}}$ is replaced with $a[_{\xi_{p_1,\dots,p_{m-1}(p^2_m)}}$.

The lemma is a straightforward consequence of these remarks and the definition of a congruent sequence. □

Lemma A.5
If $(p_1,\dots,p_k)$ and $(q_1,\dots,q_\ell)$ are both $\mathscr{G}$-congruent, then

$$(p_1,\dots,p_k,q_1,\dots,q_\ell)$$

is $\mathscr{G}$-congruent.

Proof. According to the previous lemma, $\xi_{p_1,\dots,p_k}(q_1,\dots,q_\ell)$ is $\xi_{p_1,\dots,p_k}\big(\mathscr{G}\big)$-congruent. Coming back to the definition this proves that

$$\xi_{\xi_{p_1,\dots,p_k}(q_1,\dots,q_{\ell-1})} \circ \xi_{p_1,\dots,p_k}(q_\ell)$$

is

$$\xi_{\xi_{p_1,\dots,p_k}(q_1,\dots,q_{\ell-1})} \circ \xi_{p_1,\dots,p_k}\big(\mathscr{G}\big)\text{-congruent.}$$

In lemma A.1 on page 35 we have moreover proved that

$$\xi_{p_1,\dots,p_k,q_1,\dots,q_{\ell-1}} = \xi_{\xi_{p_1,\dots,p_k}(q_1,\dots,q_{\ell-1})} \circ \xi_{p_1,\dots,p_k}.$$

This identity applied to the above statement shows that $(p_1,\dots,p_k,q_1,\dots,q_\ell)$ satisfies the definition of a $\mathscr{G}$-congruent sequence. □

Proposition A.6
If $(p_1,\dots,p_k)$ and $(q_1,\dots,q_\ell)$ are both maximal $\mathscr{G}$-congruent, then

$$\xi_{p_1,\dots,p_k}\big(\mathscr{G}\big) \equiv \xi_{q_1,\dots,q_\ell}\big(\mathscr{G}\big) \equiv \chi\big(\mathscr{G}\big).$$

Proof. From the previous lemma, $(p_1,\dots,p_k,q_1,\dots,q_\ell)$ is $\mathscr{G}$-congruent. Since p is maximal, $\xi_{p_1,\dots,p_k,q_1,\dots,q_\ell} = \xi_{p_1,\dots,p_k}$. In the same way $\xi_{q_1,\dots,q_\ell,p_1,\dots,p_k} = \xi_{q_1,\dots,q_\ell}$. We have seen moreover in a previous lemma that

$$\xi_{p_1,\dots,p_k,q_1,\dots,q_\ell} \equiv \xi_{q_1,\dots,q_\ell,p_1,\dots,p_k}.$$

This proves that $\xi_{p_1,\dots,p_k} \equiv \xi_{q_1,\dots,q_\ell}$.

We see from the definition of χ (see Definition 6.4 on page 16) that there is some maximal $\mathscr{G}$-congruent sequence $r_1,\dots,r_m$ such that $\chi\big(\mathscr{G}\big) = \xi_{r_1,\dots,r_m}\big(\mathscr{G}\big)$. Therefore

$$\chi\big(\mathscr{G}\big) \equiv \xi_{p_1,\dots,p_k}\big(\mathscr{G}\big) \equiv \xi_{q_1,\dots,q_\ell}\big(\mathscr{G}\big).$$

□

Proposition A.7
For any $\mathscr{G}, \mathscr{G}' \in \mathfrak{G}$,

$$\chi\big(\chi(\mathscr{G}) \boxplus \mathscr{G}'\big) = \chi\big(\mathscr{G} \boxplus \mathscr{G}'\big).$$

Consequently, for any $\mathscr{G}, \mathscr{G}', \mathscr{G}'' \in \mathfrak{G}$,

$$\chi\Big(\chi\big(\mathscr{G} \boxplus \mathscr{G}'\big) \boxplus \mathscr{G}''\Big) = \chi\Big(\mathscr{G} \boxplus \mathscr{G}' \boxplus \mathscr{G}''\Big).$$

Proof. Let us assume that $\mathscr{G}$, $\mathscr{G}'$ and $\chi(\mathscr{G})$ use disjoint label sets, so that

$$\chi\big(\chi\big(\mathscr{G}\big) \boxplus \mathscr{G}'\big) \equiv \chi\big(\chi(\mathscr{G}) + \mathscr{G}'\big),$$
$$\chi\big(\mathscr{G} \boxplus \mathscr{G}'\big) \equiv \chi\big(\mathscr{G} + \mathscr{G}'\big).$$

Let $p_1, \dots, p_k$ be some maximal $\mathscr{G}$-congruent sequence. It is also $\big(\mathscr{G} + \mathscr{G}'\big)$-congruent, and since label sets are disjoint,

$$\xi_{p_1,\dots,p_k}\big(\mathscr{G}\big) + \mathscr{G}' = \xi_{p_1,\dots,p_k}\big(\mathscr{G} + \mathscr{G}'\big).$$

Let us continue the sequence $p_1, \dots, p_k$ to form a maximal $\big(\mathscr{G}+\mathscr{G}'\big)$-congruent sequence $p_1, \dots, p_\ell$. Let $(q_{k+1}, \dots, q_\ell)$ be defined as

$$q_m = \xi_{p_1,\dots,p_k}(p_{k+m}).$$

We see from the definitions that $(q_{k+1}, \dots, q_\ell)$ is a maximal $\xi_{p_1,\dots,p_k}\big(\mathscr{G} + \mathscr{G}'\big)$-congruent sequence, and therefore a maximal $\Big(\xi_{p_1,\dots,p_k}(\mathscr{G}) + \mathscr{G}'\Big)$-congruent sequence. Consequently

$$\begin{aligned}\chi\Big(\chi\big(\mathscr{G}\big) + \mathscr{G}'\Big) &\equiv \xi_{q_{k+1},\dots,q_\ell}\Big(\xi_{p_1,\dots,p_k}\big(\mathscr{G}\big) + \mathscr{G}'\Big) \\ &= \xi_{q_{k+1},\dots,q_\ell} \circ \xi_{p_1,\dots,p_k}\big(\mathscr{G} + \mathscr{G}'\big) = \xi_{\xi_{p_1,\dots,p_k}(p_{k+1},\dots,p_\ell)} \circ \xi_{p_1,\dots,p_k}\big(\mathscr{G} + \mathscr{G}'\big) \\ &= \xi_{p_1,\dots,p_\ell}\big(\mathscr{G} + \mathscr{G}'\big) \equiv \chi\Big(\mathscr{G} + \mathscr{G}'\Big),\end{aligned}$$

proving the proposition. □

Proof of lemma 8.1 on page 19. Let π be the projection of $\mathfrak{G}$ on $\mathfrak{G}/\!\equiv$.

From the law of large numbers, we have that, for all $\mathscr{G} \in \mathfrak{G}$,

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(G_i \equiv \mathscr{G}) \underset{n\to\infty}{\longrightarrow} \mathbb{G}(\pi(\mathscr{G})).$$

Let us now remark that $\boxplus_{i=1}^{n} n^{-1} G_i = \boxplus_{\overline{\mathscr{G}} \in \mathfrak{G}/\equiv} \boxplus_{\substack{i \\ G_i \in \overline{\mathscr{G}}}} n^{-1} G_i$. Thus

$$\frac{1}{n}\chi\left(\boxplus_{i=1}^{n} G_i\right) = \chi\left(\boxplus_{\overline{\mathscr{G}} \in \mathfrak{G}/\equiv} \chi\left(\boxplus_{\substack{i \\ G_i \in \overline{\mathscr{G}}}} n^{-1} G_i\right)\right)$$

$$= \chi\left(\boxplus_{\overline{\mathscr{G}}\in\mathfrak{G}/\equiv}\left(\sum_{i=1}^{n} n^{-1}\mathbb{1}(G_i\in\overline{\mathscr{G}})\right)\chi(\overline{\mathscr{G}})\right)$$
$$= \chi\left(\boxplus_{\overline{\mathscr{G}}\in\mathfrak{G}/\equiv}\left(\sum_{i=1}^{n} n^{-1}\mathbb{1}(G_i\in\overline{\mathscr{G}})\right)\overline{\mathscr{G}}\right).$$

We used here Proposition A.7 on the preceding page and the fact that for any $a, b \in \mathbb{R}_+$,

$$\chi\big[(a\mathscr{G})\boxplus(b\mathscr{G})\big] = (a+b)\,\chi(\mathscr{G}),$$

which comes from the following reasoning: Suppose that

$$\{1,\dots,d\} = \{i\,;\ \mathscr{G}([_iS^*) > 0\},$$

and let $p_i = (2i, 2i-1)$. Since each p_i is $(a\mathscr{G})\boxplus(b\mathscr{G})$-congruent, $(p_1,\dots,p_d)$ is also $(a\mathscr{G})\boxplus(b\mathscr{G})$-congruent, from lemma A.5 on page 37. It is quite straightforward to see that

$$\xi_{p_1,\dots,p_d}\big[(a\mathscr{G})\boxplus(b\mathscr{G})\big] \equiv (a+b)\mathscr{G}.$$

This implies that

$$\chi\big[(a\mathscr{G})\boxplus(b\mathscr{G})\big] = \chi\circ\xi_{p_1,\dots,p_d}\big[(a\mathscr{G})\boxplus(b\mathscr{G})\big] = \chi\big[(a+b)\mathscr{G}\big] = (a+b)\,\chi(\mathscr{G}).$$

To take the limit inside χ, we need to prove that χ is continuous in a suitable sense. Actually, $\mathcal{G}\mapsto\chi(\mathcal{G})$ is continuous on sets of fixed support, and this is what is required to conclude.

Indeed, for any sequence $(\mathscr{G}_i)$ with fixed support for n large enough, there is a fixed label map f (depending on the support) such that for n large enough $\chi(\mathscr{G}_i) = f(\mathscr{G}_i)$, and the result follows from the fact that $\mathscr{G}\mapsto f(\mathscr{G})$ is continuous; since $f(\mathscr{G})(A) = \mathscr{G}(f^{-1}(A))$.

Consequently

$$\lim_{n\to\infty}\frac{1}{n}\,\chi\left(\boxplus_{i=1}^{n} G_i\right) = \chi\left(\boxplus_{\overline{\mathscr{G}}\in\mathfrak{G}/\equiv}\lim_{n\to\infty}\left(\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}\big(G_i\in\overline{\mathscr{G}}\big)\right)\overline{\mathscr{G}}\right)$$
$$= \chi\left(\boxplus_{\overline{\mathscr{G}}\in\mathfrak{G}/\equiv}\mathbb{G}(\overline{\mathscr{G}})\overline{\mathscr{G}}\right) = \chi\left(\boxplus_{\overline{\mathscr{G}}\in\mathfrak{G}/\equiv}\mathbb{G}(\overline{\mathscr{G}})\chi(\overline{\mathscr{G}})\right)$$
$$= \chi\left(\boxplus_{\overline{\mathscr{G}}\in\mathfrak{G}/\equiv}\ \boxplus_{\mathscr{G}\in\overline{\mathscr{G}}}\mathbb{G}(\mathscr{G})\chi(\mathscr{G})\right) = \chi\left(\boxplus_{\mathscr{G}\in\mathfrak{G}}\mathbb{G}(\mathscr{G})\chi(\mathscr{G})\right)$$
$$= \chi\left(\boxplus_{\mathscr{G}\in\mathfrak{G}}\mathbb{G}(\mathscr{G})\mathscr{G}\right) = \oint\mathscr{G}\,\mathrm{d}\mathbb{G}(\mathscr{G}).\qquad\square$$

Appendix B: Language produced by a Toric Grammar

In this appendix, we make a deterministic study of the language produced by a toric grammar $\mathscr{G} \in \beta^*(\mathfrak{T})$. More precisely, we are interested in the support of the distribution $\mathbb{T}_{\mathscr{G}}$ of the final state of the production process.

Lemma B.1
Let $\mathscr{T} \in \mathfrak{T}$ be some text and $\mathscr{G} \in \beta^*(\mathscr{T})$ be some grammar obtained by splitting this text a finite number of times. The number of splits performed can be read in $\mathscr{G}$ and is equal to

$$n = \sum_{i=1}^{+\infty} \mathscr{G}\big(\,]_i S^*\big).$$

Let us put $\overline{\alpha}(\mathscr{G}) = \alpha^n(\mathscr{G})$. Then, $\mathscr{T} \in \overline{\alpha}(\mathscr{G}) \subset \mathfrak{T}$, moreover $\overline{\alpha}(\mathscr{G}) = \operatorname{supp}(\mathbb{T}_{\mathscr{G}})$.

Proof. The grammar $\mathscr{G}$ is obtained by making a succession of splits. Each of those splits add one $[_i$ and one $]_i$ to the grammar, whereas in the original text there are no $[_i$ nor $]_i$, except for the $[_0$ at the beginning of each sentence. Since application of an element of $\mathfrak{F}$ does not change the number of such symbols, they may be used to count the number of splits performed.

Let us take then a sequence of toric grammars $\mathscr{T} = \mathscr{G}_0, \dots, \mathscr{G}_n = \mathscr{G}$, such that $\mathscr{G}_k \in \beta(\mathscr{G}_{k-1})$. From lemma 4.2 on page 10, there is a sequence $f_1, \dots, f_n \in \mathfrak{F}$ such that $f_k\big(\mathscr{G}_{k-1}\big) \in \alpha\big(\mathscr{G}_k\big)$. Let us prove by induction that for any $k = 0, \dots, n$,

$$f_k \circ \cdots \circ f_1\big(\mathscr{T}\big) \in \alpha^k\big(\mathscr{G}_k\big).$$

Indeed, this is true for $k = 0$, since $\mathscr{G}_0 = \mathscr{T}$. Moreover, assuming that the assertion holds for $k-1$, we deduce that

$$f_k \circ \cdots \circ f_1\big(\mathscr{T}\big) \in f_k\Big(\alpha^{k-1}\big(\mathscr{G}_{k-1}\big)\Big) \subset \alpha^{k-1}\Big(f_k\big(\mathscr{G}_{k-1}\big)\Big) \subset \alpha^k\big(\mathscr{G}_k\big).$$

showing that if the assertion holds for k, it also holds for $k+1$. For $k = n$, we obtain that

$$f_n \circ \cdots \circ f_1\big(\mathscr{T}\big) \in \alpha^n\big(\mathscr{G}_n\big).$$

As $f_n \circ \cdots \circ f_1\big(\mathscr{T}\big) = \mathscr{T}$, since $\mathscr{T}$ is a text, and $\mathscr{G}_n = \mathscr{G}$, we get that $\mathscr{T} \in \alpha^n\big(\mathscr{G}\big)$.

Let us consider now $\mathscr{G}' \in \overline{\alpha}(\mathscr{G})$. Let $(\mathscr{G} = \mathscr{G}_0, \dots, \mathscr{G}_n = \mathscr{G}')$ the chain of grammars leading to $\mathscr{G}'$. Then for any $k = 0, \dots, n$,

$$\sum_{i=1}^{+\infty} \mathscr{G}_k\big(\,]_i S^*\big) = n - k,$$

since $\mathscr{G}_k \in \alpha(\mathscr{G}_{k-1})$ and each merge takes away one $[_i$ and one $]_i$. This implies that $\sum_{i=1}^{+\infty} \mathscr{G}'\big(\,]_i S^*\big) = 0$, and thus $\mathscr{G}' \in \mathfrak{T}$.

Note that, as remarked above, repeated merges may create elements of the type $[_i\, a\,]_i b$. However, this will not happen if n successful merges can be performed. Indeed in the case when expressions of the form $[_i\, a\,]_i b$ remain unmatched during the merge process, we will get $\alpha\big(\mathscr{G}_k\big) = \varnothing$ for some $k < n$. □

References

[1] Baker, J.K. (1979). Trainable grammars for speech recognition. *The Journal of the Acoustical Society of America*, **65**(S1) S132–S132.

[2] Chi, Z. and Geman, S. (1998). Estimation of probabilistic context-free grammars. *Computational Linguistics*, **24**(2) 299–305. MIT Press.

[3] Chi, Z. (1999). Statistical properties of probabilistic context-free grammars. *Computational Linguistics*, **25**(1) 131–160. MIT Press.

[4] Chomsky, N. (1956). Three Models for the Description of Language. *IRE Transactions on Information Theory*

[5] Chomsky, N. (1957). *Syntactic Structures*. Mouton & Co.

[6] Chomsky, N. (1965). *Aspects of the Theory of Syntax.* MIT Press.

[7] Chomsky, N. (1995). *The Minimalist Program.* MIT Press.

[8] Cohen, S. B. and Smith, N. A. (2012). Empirical Risk Minimization for Probabilistic Grammars: Sample Complexity and Hardness of Learning. *Computational Linguistics*. **38**(3) 479–526.

[9] Della Pietra, S., Della Pietra, V., Gillett, J., Lafferty, J., Printz, H. and Ureš, L.. Inference and estimation of a long-range trigram model. *Grammatical Inference and Applications*, 78–92. Springer.

[10] Lari, K. and Young, S. (1990). The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer speech & language*, **4**(1) 35–56. Elsevier.

[11] Norris, J. R. (1998) *Markov Chains*. Cambridge University Press.

[12] Roark, B. (2001). Probabilistic Top-Down Parsing and Language Modeling. *Computational Linguistics*. **27**(2) 249–276.

[13] Sakakibara, y. (1990). Learning context-free grammars from structural data in polynomial time. *Theoretical Computer Science*, **76** (2) 223–242. Elsevier.

[14] Stabler, E. (2009) Mathematics of language learning. *Histoire, Épistémologie, Langage* **31**(1) 127–145.

[15] Tan, M., Zhou, W., Zheng, L. and Wang, S. (2012) A Scalable Distributed Syntactic, Semantic, and Lexical Language Model. *Computational Linguistics*. **38**(3) 631–671.